



Tutorial para desarrolladores de Slicer: Programación en Slicer

Sonia Pujol, Ph.D.

Profesora adjunta de radiología

Directora de formación y educación en 3D Slicer

Hospital de mujeres de Brigham

Facultad de Medicina de Harvard

Steve Pieper, Ph.D.

Arquitecto jefe de 3D Slicer

Isomics Inc.

Objetivo del tutorial



```
def threshold(t):  
    n=getNode('T2')  
    a=array('T2')  
    a[a<t]=0  
    arrayFromVolumeModified(n)  
    print('Thresholding done')
```



```
b=qt.QPushButton('Toggle')  
b.connect('clicked()',toggle)  
b.styleSheet = "font-size: 24pt; color:  
aqua; margin: 20px"  
b.show()
```

Este tutorial es una introducción a la consola de Python y al widget Qt de la versión 5 de 3D Slicer

Esquema del Tutorial



Parte 1: Visión general de los módulos de 3D Slicer



Parte 2: Familiarizarse con el entorno Python en 3D Slicer

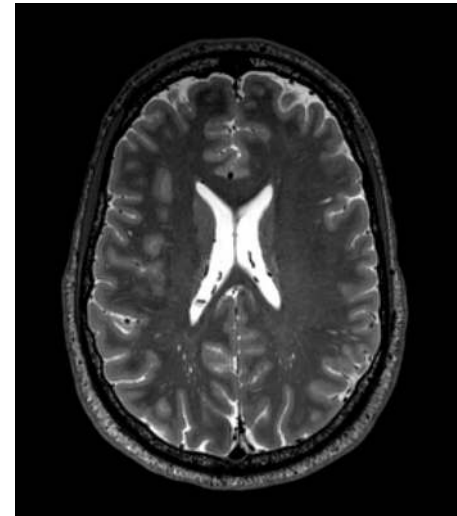
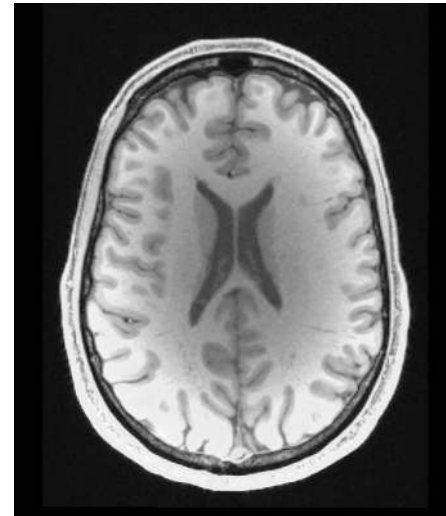
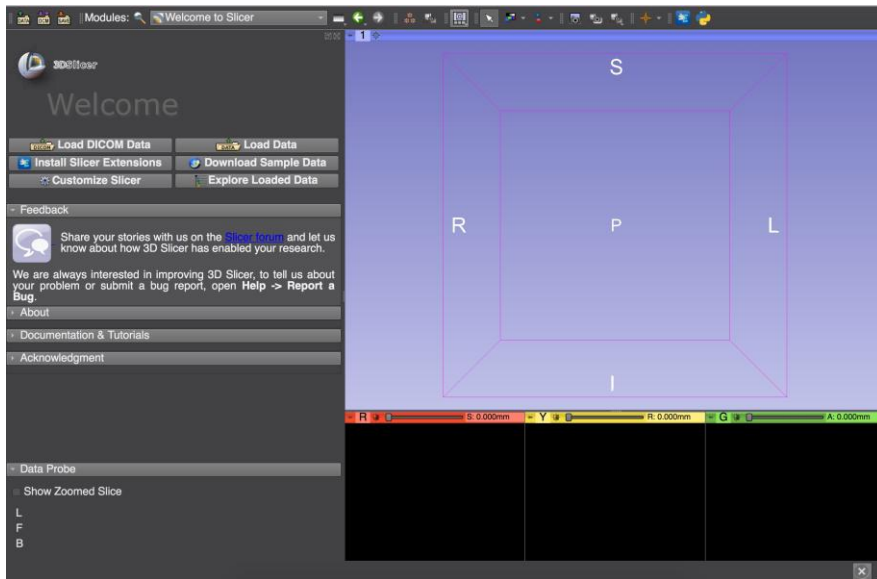


Parte 3: Familiarizarse con el conjunto de herramientas de widgets Qt en 3D Slicer

Descargo de responsabilidad

- 3D Slicer es una aplicación de software libre de código abierto distribuida bajo una licencia de estilo BSD.
- El software no cuenta con la aprobación de la FDA ni el mercado CE y es para uso exclusivo en investigación.

Materiales didácticos



SlicerProgrammingTutorialData.zip
[link](#)

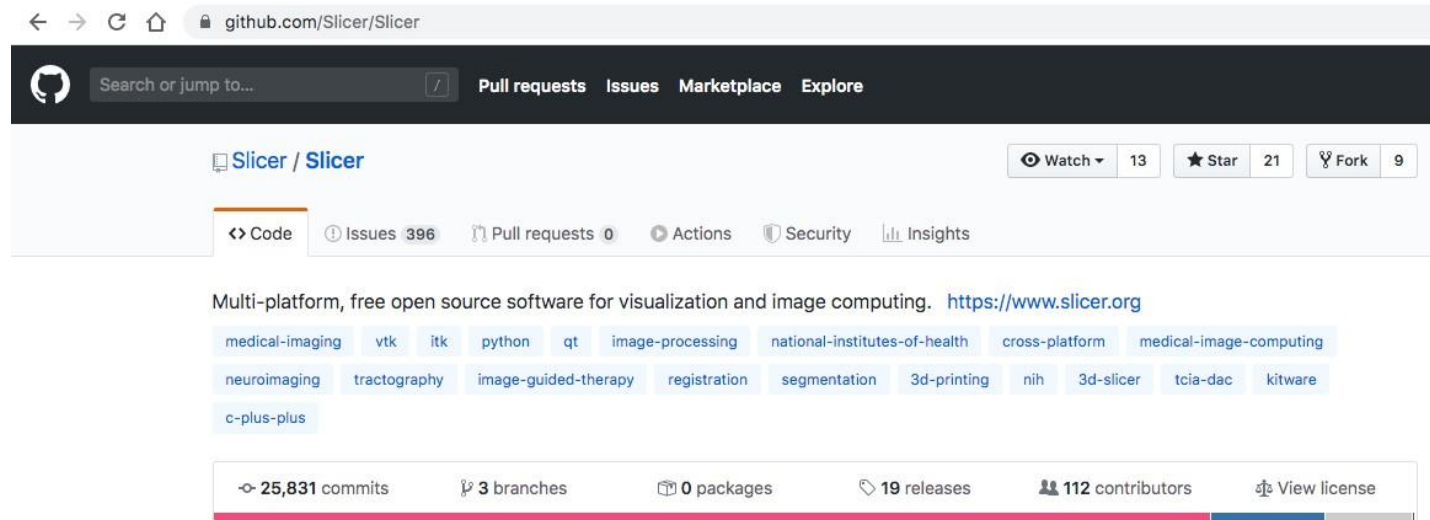
3D Slicer versión 5.10

Parte 1

Visión general de los módulos de Slicer



3D Slicer



- 3D Slicer es una plataforma de código abierto para el análisis y la visualización de datos de imágenes médicas.
- 3D Slicer se compila y se prueba a diario en plataformas Windows, Mac y Linux
- El código fuente está disponible gratuitamente en GitHub en <http://github.com/Slicer/Slicer>

Módulos Slicer

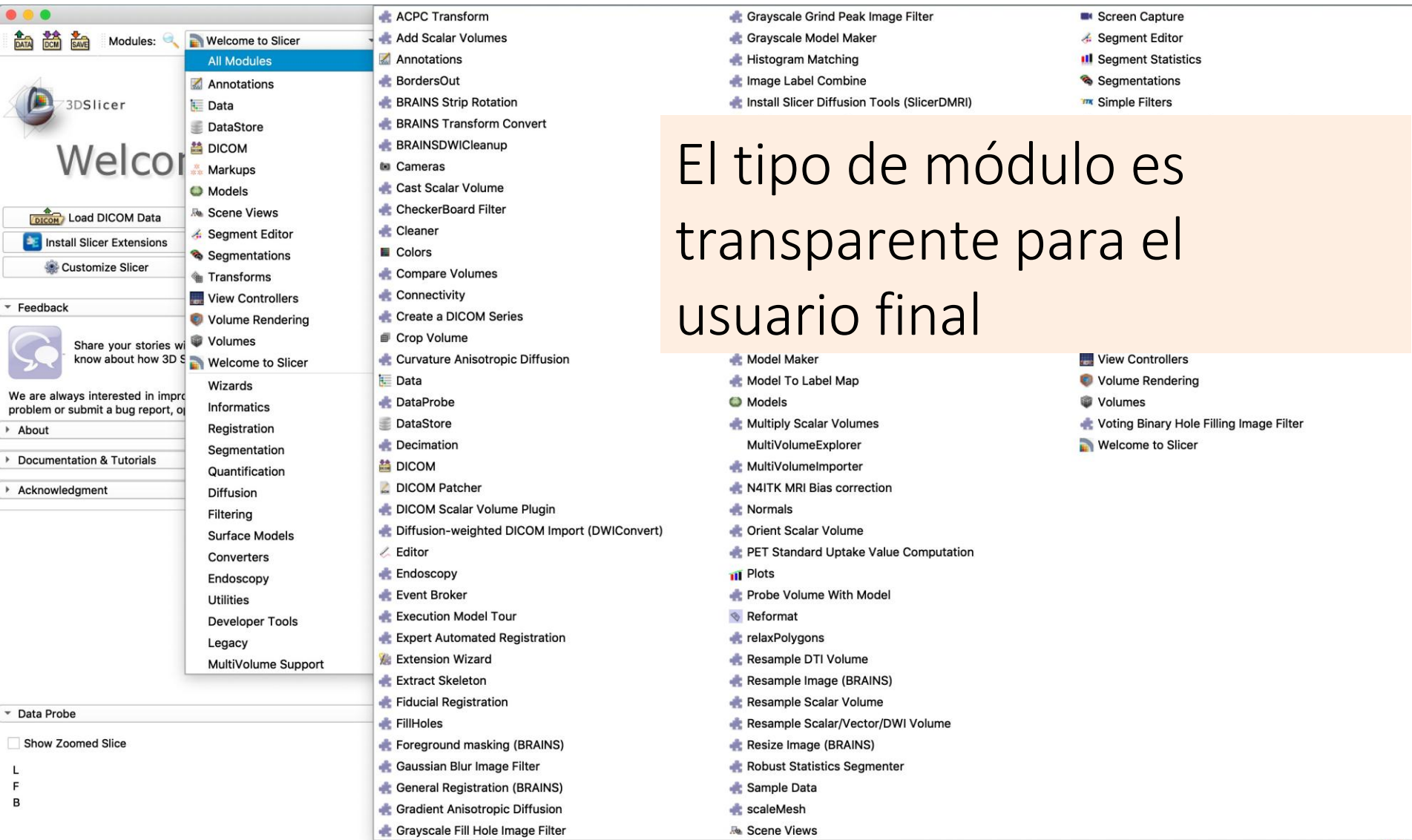
3D Slicer admite tres tipos de módulos:

- **Interfaz de línea de comandos (CLI):** ejecutable independiente con argumentos de entrada/salida limitados.
- **Módulos cargables (C++ Plugins):** optimizados para cálculos pesados.

Es el enfoque de este tutorial

- **Módulos con script (Python):** recomendados para la creación rápida de prototipos y el desarrollo de flujo de trabajo.

Módulos Slicer



The screenshot displays the Slicer application interface. On the left, the 'Modules' menu is open, showing a list of categories and sub-modules. The main window displays a list of available modules, organized into three columns. A large orange text box is overlaid on the right side of the module list.

Modules Menu:

- Welcome to Slicer
- All Modules
- Annotations
- Data
- DataStore
- DICOM
- Markups
- Models
- Scene Views
- Segment Editor
- Segmentations
- Transforms
- View Controllers
- Volume Rendering
- Volumes
- Welcome to Slicer
- Wizards
- Informatics
- Registration
- Segmentation
- Quantification
- Diffusion
- Filtering
- Surface Models
- Converters
- Endoscopy
- Utilities
- Developer Tools
- Legacy
- MultiVolume Support

Available Modules:

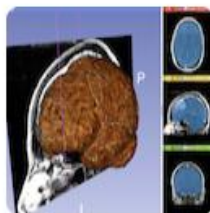
- ACPC Transform
- Add Scalar Volumes
- Annotations
- BordersOut
- BRAINS Strip Rotation
- BRAINS Transform Convert
- BRAINS DWICleanup
- Cameras
- Cast Scalar Volume
- CheckerBoard Filter
- Cleaner
- Colors
- Compare Volumes
- Connectivity
- Create a DICOM Series
- Crop Volume
- Curvature Anisotropic Diffusion
- Data
- DataProbe
- DataStore
- Decimation
- DICOM
- DICOM Patcher
- DICOM Scalar Volume Plugin
- Diffusion-weighted DICOM Import (DWICConvert)
- Editor
- Endoscopy
- Event Broker
- Execution Model Tour
- Expert Automated Registration
- Extension Wizard
- Extract Skeleton
- Fiducial Registration
- FillHoles
- Foreground masking (BRAINS)
- Gaussian Blur Image Filter
- General Registration (BRAINS)
- Gradient Anisotropic Diffusion
- Grayscale Fill Hole Image Filter
- Grayscale Grind Peak Image Filter
- Grayscale Model Maker
- Histogram Matching
- Image Label Combine
- Install Slicer Diffusion Tools (SlicerDMRI)
- Model Maker
- Model To Label Map
- Models
- Multiply Scalar Volumes
- MultiVolumeExplorer
- MultiVolumeImporter
- N4ITK MRI Bias correction
- Normals
- Orient Scalar Volume
- PET Standard Uptake Value Computation
- Plots
- Probe Volume With Model
- Reformat
- relaxPolygons
- Resample DTI Volume
- Resample Image (BRAINS)
- Resample Scalar Volume
- Resample Scalar/Vector/DWI Volume
- Resize Image (BRAINS)
- Robust Statistics Segmenter
- Sample Data
- scaleMesh
- Scene Views
- Screen Capture
- Segment Editor
- Segment Statistics
- Segmentations
- Simple Filters
- View Controllers
- Volume Rendering
- Volumes
- Voting Binary Hole Filling Image Filter
- Welcome to Slicer

Text Box:

El tipo de módulo es transparente para el usuario final

Extensiones Slicer

Una extensión Slicer es un paquete que incluye uno o varios módulos de Slicer



SwissSkullStripper
Bill Lorensen (Noware...)

★★★★★ (0)

INSTALL



PETTumorSegmenta...
Christian Bauer (Univ...)

★★★★★ (0)

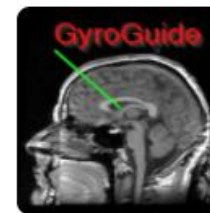
INSTALL



SlicerOpenIGTLink
Junichi Tokuda (SPL, ...)

★★★★★ (0)

INSTALL



GyroGuide
Ruifeng Chen, Luping...

★★★★★ (0)

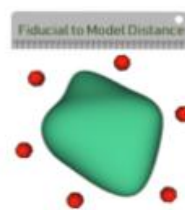
INSTALL



PET-IndiC
Ethan Ulrich (Universi...)

★★★★★ (0)

INSTALL



FiducialsToModelDi...
Jesse Reynolds (Cante...)

★★★★★ (0)

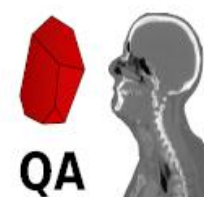
INSTALL



Slicer-Wasp
Thomas Lawson (MR...)

★★★★★ (0)

INSTALL



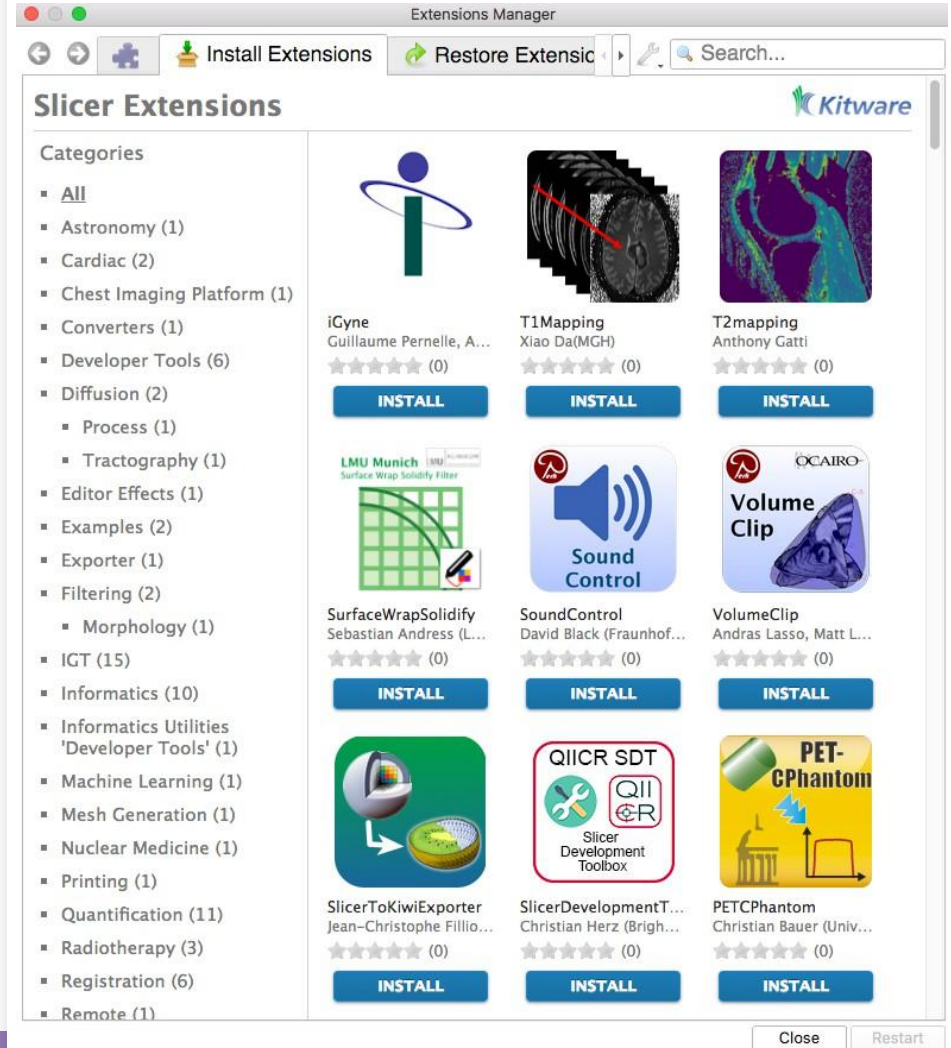
ImageCompare
Paolo Zaffino (Magna ...)

★★★★★ (0)

INSTALL

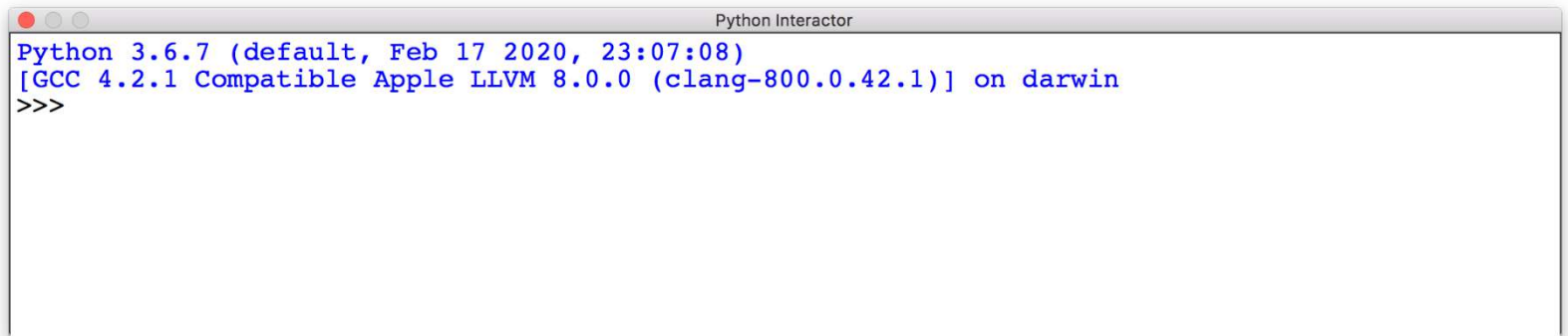
Administrador de extensiones de Slicer

- El administrador de extensiones de Slicer proporciona una plataforma de “tienda de aplicaciones” para el ecosistema de 3D Slicer
- El Extension Manager facilita la creación e instalación de extensiones de Slicer.
- La versión 5 de Slicer incluye más de 130 extensiones.



Parte 2

Familiarizarse con el entorno Python en 3D Slicer



```
Python Interactor
Python 3.6.7 (default, Feb 17 2020, 23:07:08)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.42.1)] on darwin
>>>
```

Python en Slicer

Slicer v.5.10 funciona con Python3 y un conjunto de bibliotecas estándar

	NumPy, el cual es el paquete fundamental para la computación científica con Python.
	VTK es una biblioteca de código abierto para la manipulación y visualización de datos científicos.
	ITK es una biblioteca de código abierto para el análisis de imágenes.
	CTK es una biblioteca de código abierto para el cálculo de imágenes biomédicas.
	PythonQT es un enlace de Python para Qt.
	Qt es un entorno de multiplataforma utilizado como conjunto de herramientas gráficas.

Python en Slicer

- El índice de paquetes de Python (PyPi) da acceso a más de 200.000 paquetes adicionales de Python (<http://pipy.org>).
- El comando “pip install” en Slicer permite a los desarrolladores instalar las herramientas de computación científica más comunes (por ejemplo, TensorFlow, SciPy, PyTorch, Pandas, etc.).
- Slicer puede utilizarse como kernel de Jupyter notebook
- PyCharm y otras herramientas de desarrollo de Python se pueden utilizar con Slicer



The screenshot displays the 3D Slicer 5.10.0 interface. On the left is a sidebar with a 'Bienvenido' (Welcome) section containing links for adding data, installing extensions, and downloading sample data. Below this are sections for documentation, contact information, and information about the software. The main area on the right shows a 3D view of a rectangular prism with axes labeled R (Right), L (Left), S (Superior), and I (Inferior). The prism is outlined in purple. Below the 3D view are three sliders for the axes: R (0.0000mm), A (0.0000mm), and L (0.0000mm). A 'Snipping Tool' window is open in the bottom right corner, displaying a screenshot of the 3D view and a message: 'Screenshot copied to clipboard. Automatically saved to screenshots folder. Markup and share'.

Bienvenido

Añadir datos Añadir datos DICOM

Instalar extensiones Descargar datos de muestra

Personalizar Slicer Explorar datos añadidos

Comentarios

Acerca de

Documentación y Tutoriales

Documentación

- Documentación
- Inicio rápido
- Obtenga ayuda
- Interfaz de usuario
- Botones del ratón, "Teclas de acceso rápido" y atajos de teclado
- Buscar Tutoriales

Contáctanos

- Visitar el Foro de Slicer
- Síguenos en Twitter
- Búsqueda de funciones
- Informar de un error

Sobre 3D Slicer

- Ver licencia
- Cómo citar
- Publicaciones de Slicer
- Agradecimientos

Actualizaciones

Agradecimiento

Coordenadas

Mostrar corte ampliado

L
F
B

R S L I

R: 0.0000mm A: 0.0000mm L: 0.0000mm

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Markup and share

La version 5 de Slicer integra
Python3, VTK5 and ITK5

Consola de Python en Slicer

La consola de Python esta basada en Qt y permite un acceso directo a Nodos MRML de Slicer, librerías (NumPy, VTK, ITK, CTK) y Qt.

Bienvenido

Añadir datos

Añadir datos DICOM

Instalar extensiones

Descargar datos de muestra

Personalizar Slicer

Explorar datos añadidos

Comentarios

Comparta sus historias con nosotros en el [foro de Slicer](#) y cuéntenos como 3D Slicer ha hecho posible su investigación.

Estamos siempre interesados en mejorar 3D Slicer, para contarnos su problema o enviar un informe de error, abra **Ayuda-> Informar sobre un error.**

Acerca de

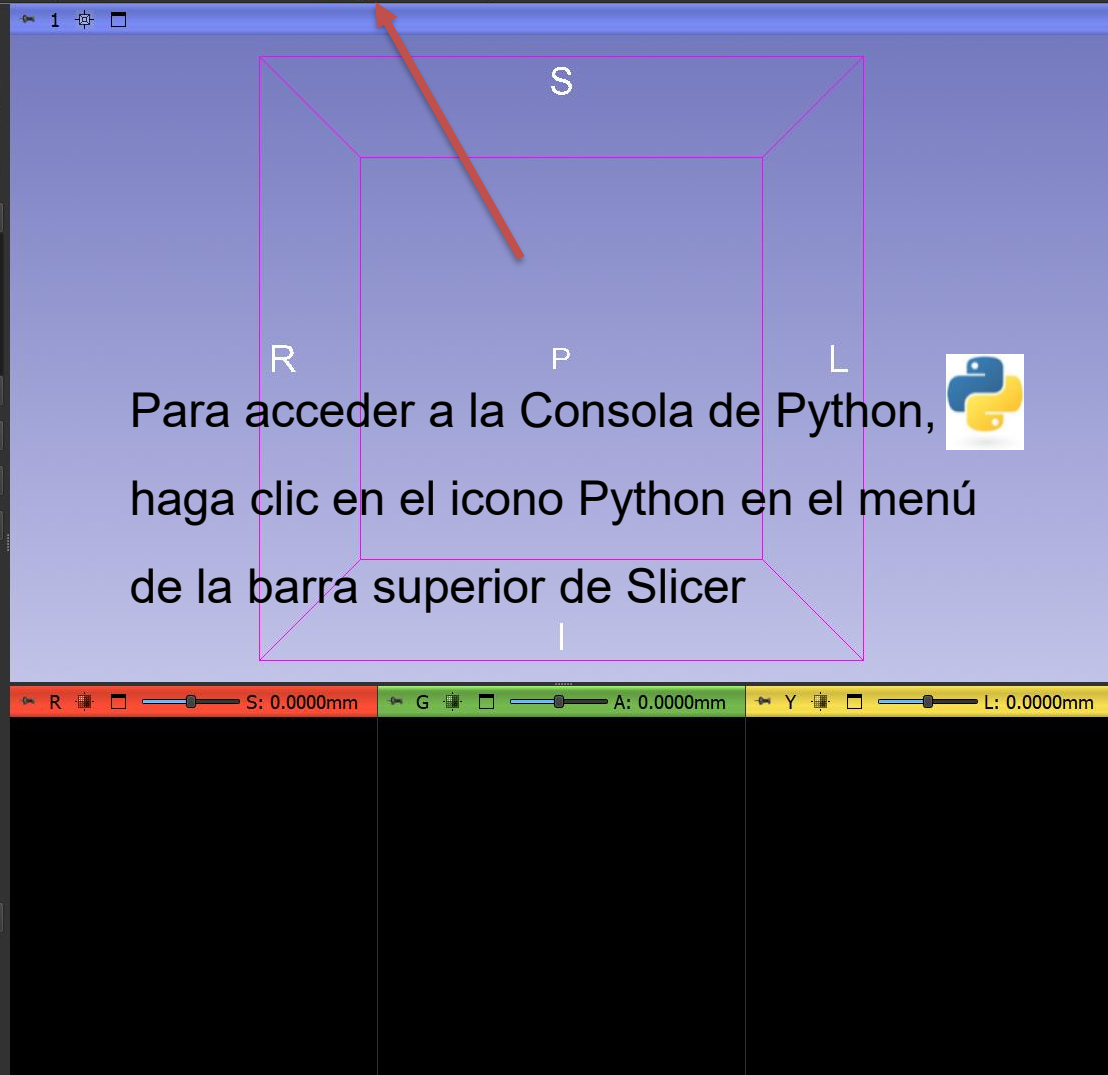
Documentación y Tutoriales

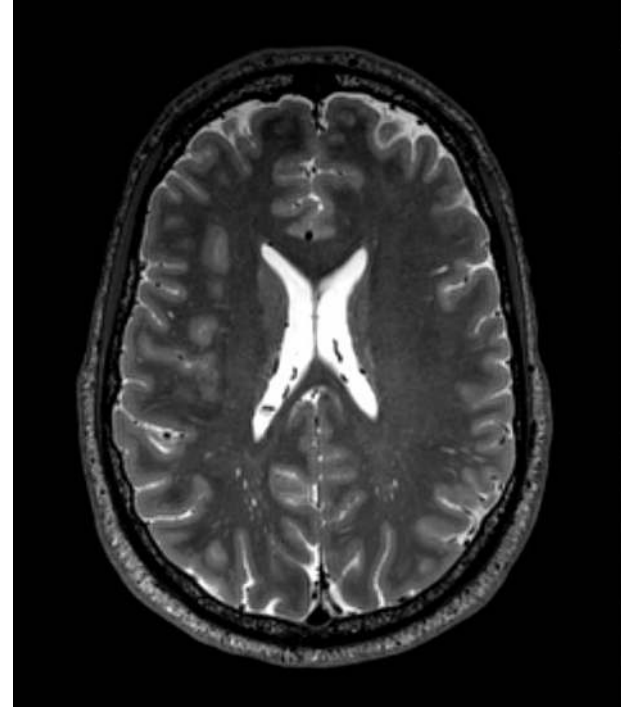
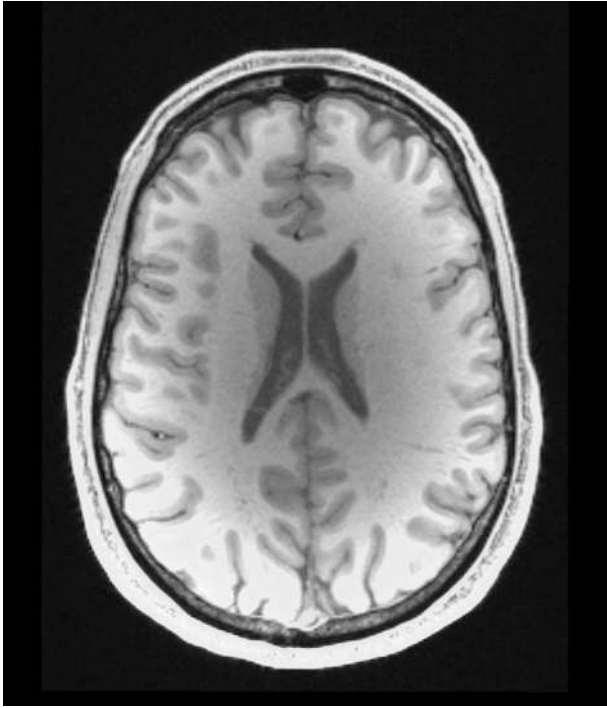
Actualizaciones

Agradecimiento

Coordenadas

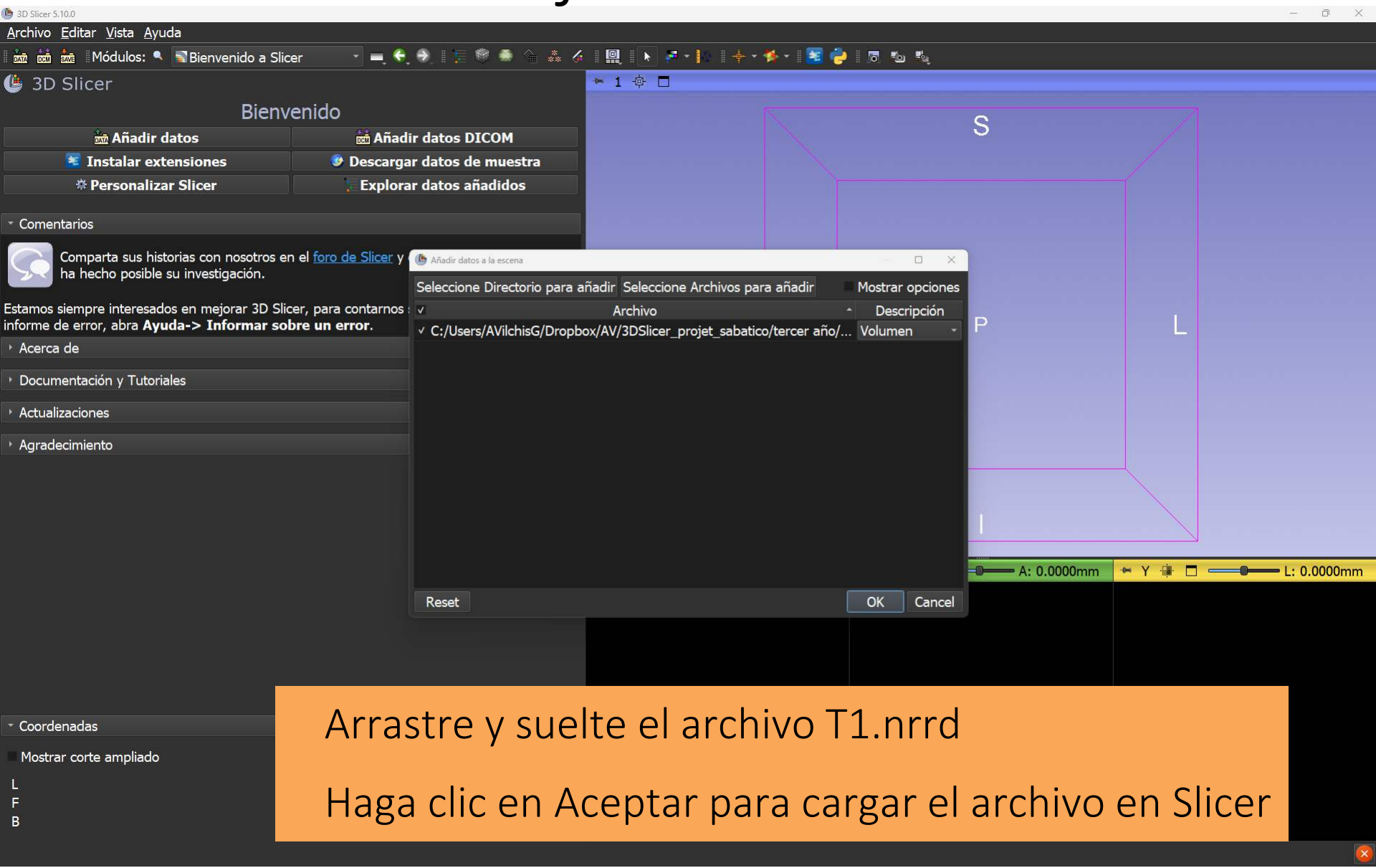
Mostrar corte ampliado

L
F
B

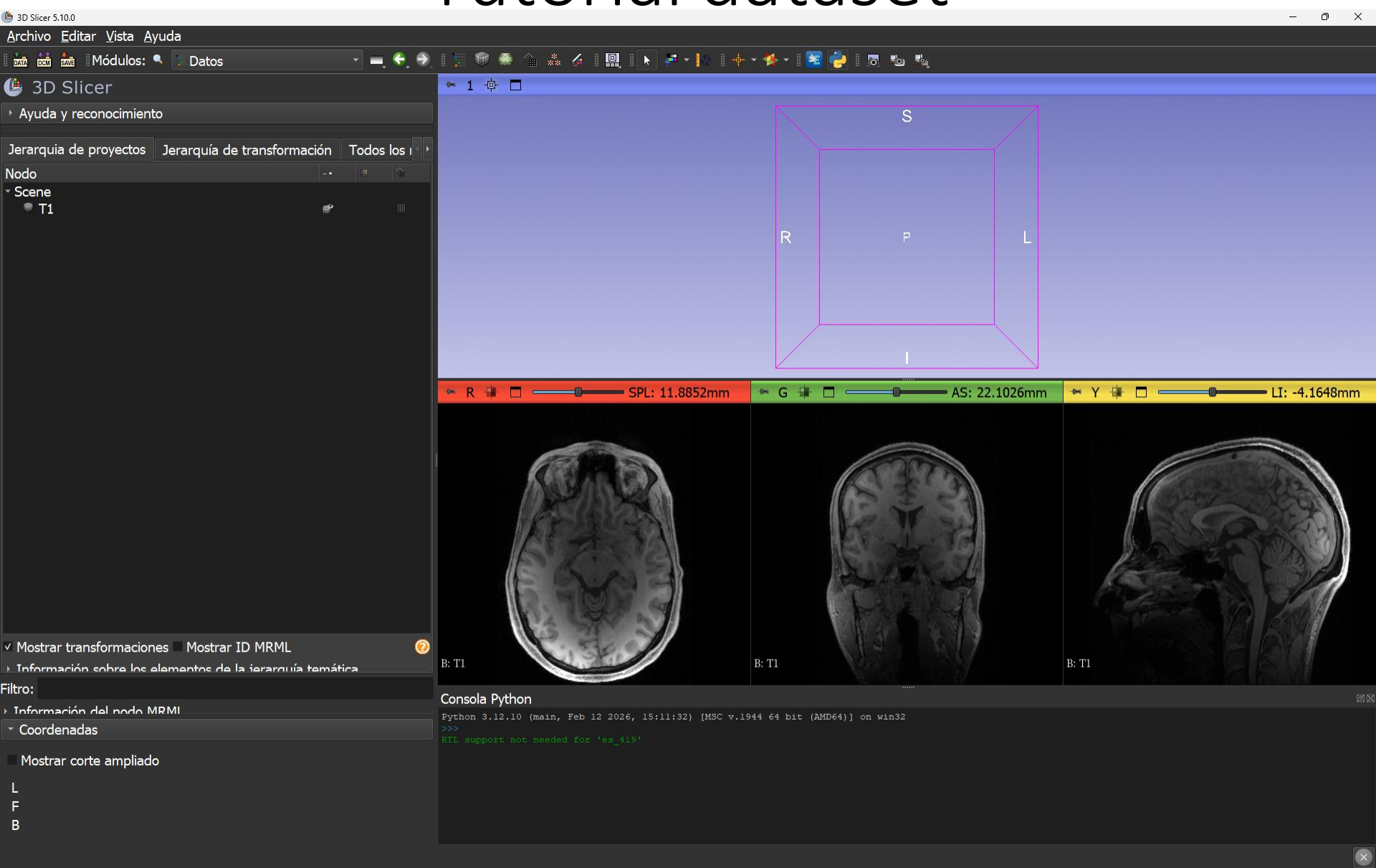


El conjunto de datos del tutorial de programación Slicer incluye una resonancia magnética ponderada en T1 y otra en T2 de un sujeto sano.

Conjunto de datos



Tutorial dataset



Panorama general

- Slicer es un software gratuito y de código abierto
- Hay miles de imágenes médicas sofisticadas disponibles en Internet que puede visualizar y analizar con 3D Slicer.

Modelo de datos de Slicer



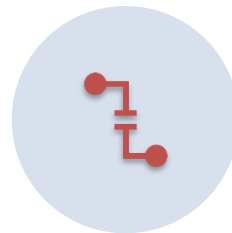
El modelo de datos de Slicer se basa en la estructura de datos de escena de Slicer.



Una escena Slicer es una colección de imágenes, anotaciones, modelos 3D, transformaciones espaciales, fiduciales y cámaras.



El lenguaje MRML (Medical Reality Markup Language) es un lenguaje basado en XML que se utiliza para serializar el contenido de la escena Slicer en disco (scene.mrml).



Cada elemento de una escena se denomina nodo MRML

Nodos MRML de Slicer: Tipos Básicos



Nodo de datos:
Almacena los datos en bruto.

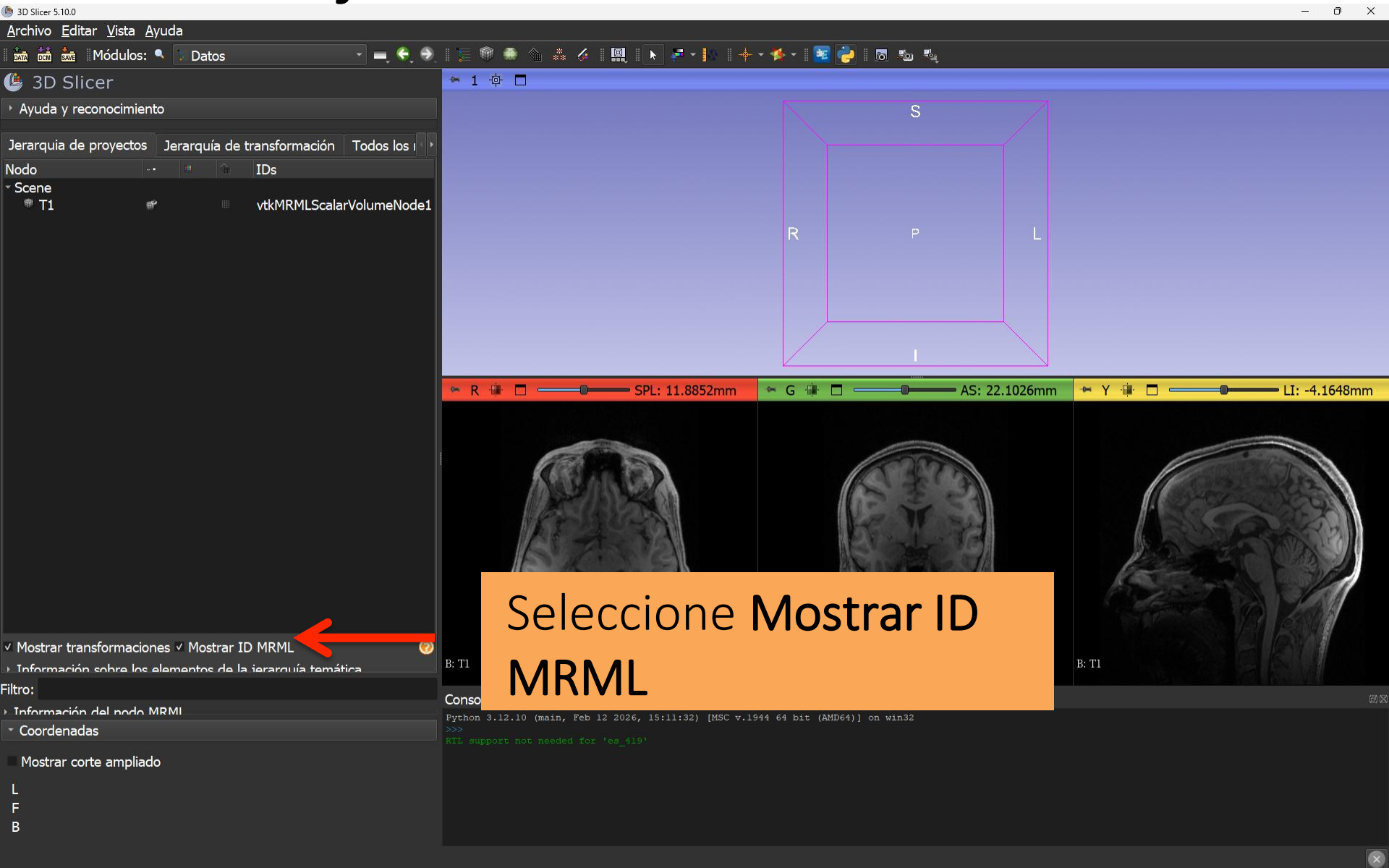


Nodo de visualización:
Describe cómo deben visualizarse los datos.

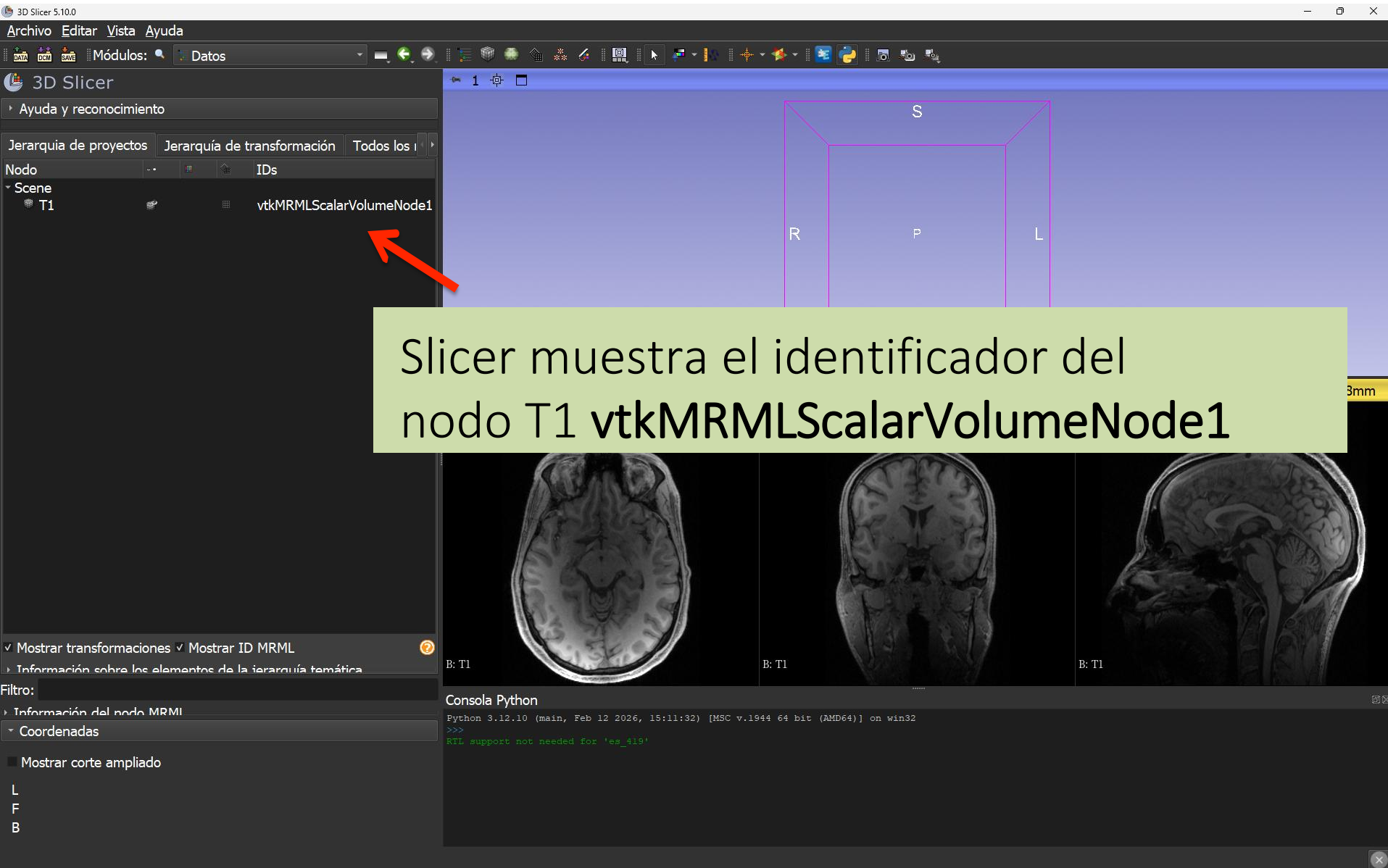


Nodo de almacenamiento:
Describe cómo deben almacenarse los datos en el disco.

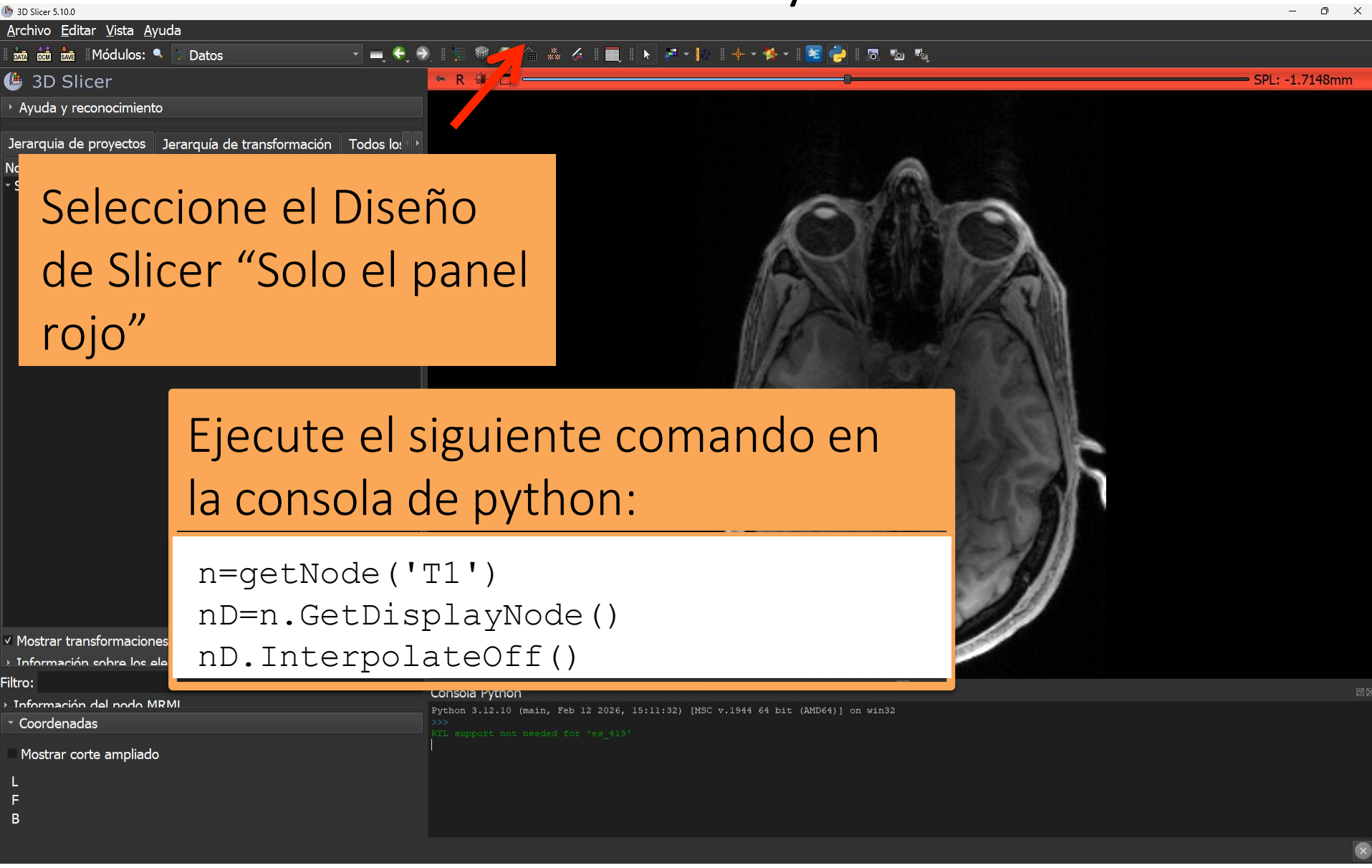
Conjunto de datos del tutorial



Slicer Data Model



Acceso a los nodos MRML desde la consola Python



Seleccione el Diseño de Slicer “Solo el panel rojo”

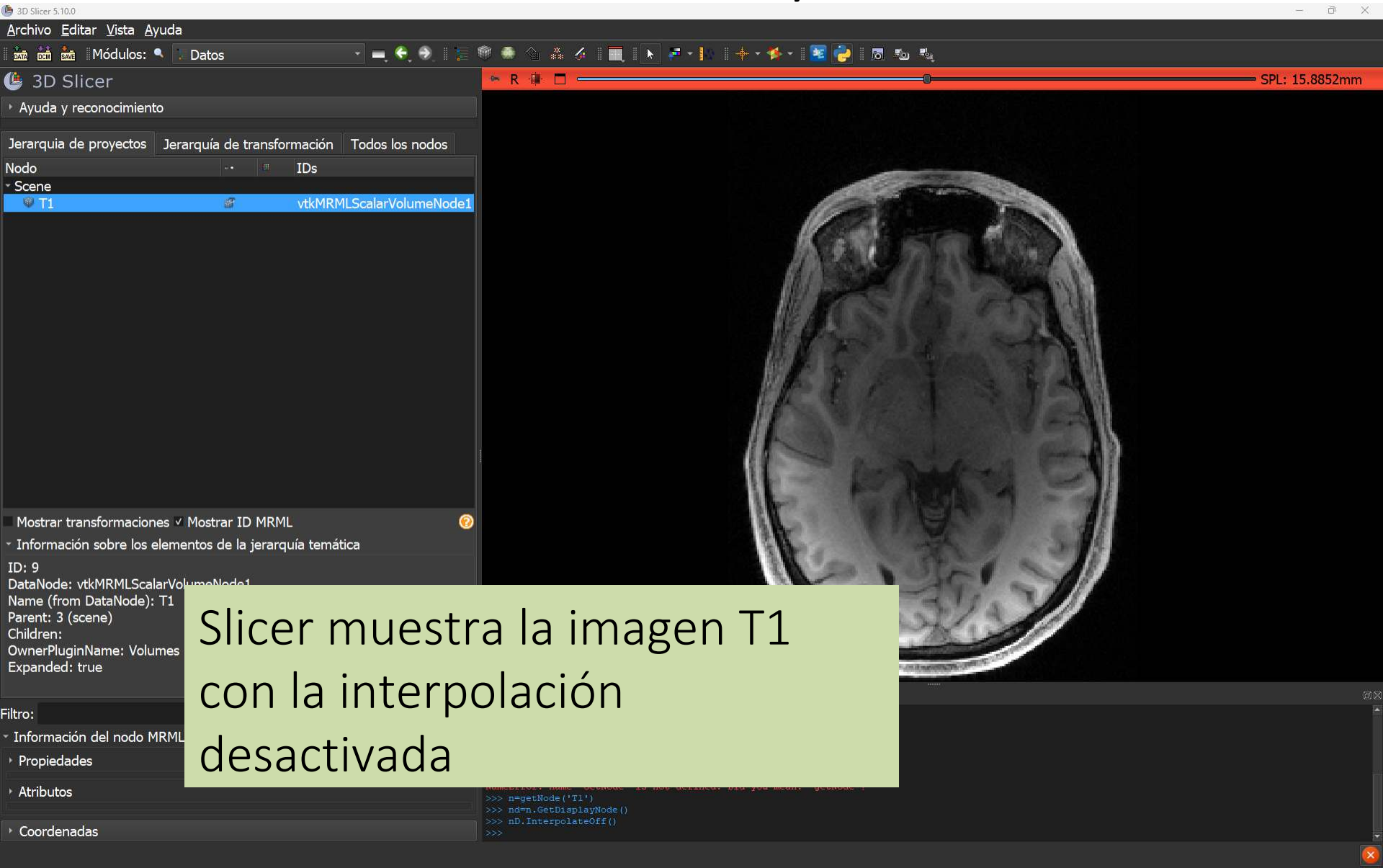
Ejecute el siguiente comando en la consola de python:

```
n=getNode('T1')
nD=n.GetDisplayNode()
nD.InterpolateOff()
```

Consola Python

```
Python 3.12.10 (main, Feb 12 2026, 15:11:32) [MSC v.1944 64 bit (AMD64)] on win32
>>>
RTL support not needed for 'es_419'
```

Acceso a los nodos MRML desde la consola Python



3D Slicer 5.10.0

Archivo Editar Vista Ayuda

Módulos: Datos

3D Slicer

Ayuda y reconocimiento

Jerarquía de proyectos Jerarquía de transformación Todos los nodos

Nodo IDs

Scene

T1 vtkMRMLScalarVolumeNode1

Mostrar transformaciones ✓ Mostrar ID MRML

Información sobre los elementos de la jerarquía temática

ID: 9
DataNode: vtkMRMLScalarVolumeNode1
Name (from DataNode): T1
Parent: 3 (scene)
Children:
OwnerPluginName: Volumes
Expanded: true

Filtro:

Información del nodo MRML

Propiedades

Atributos

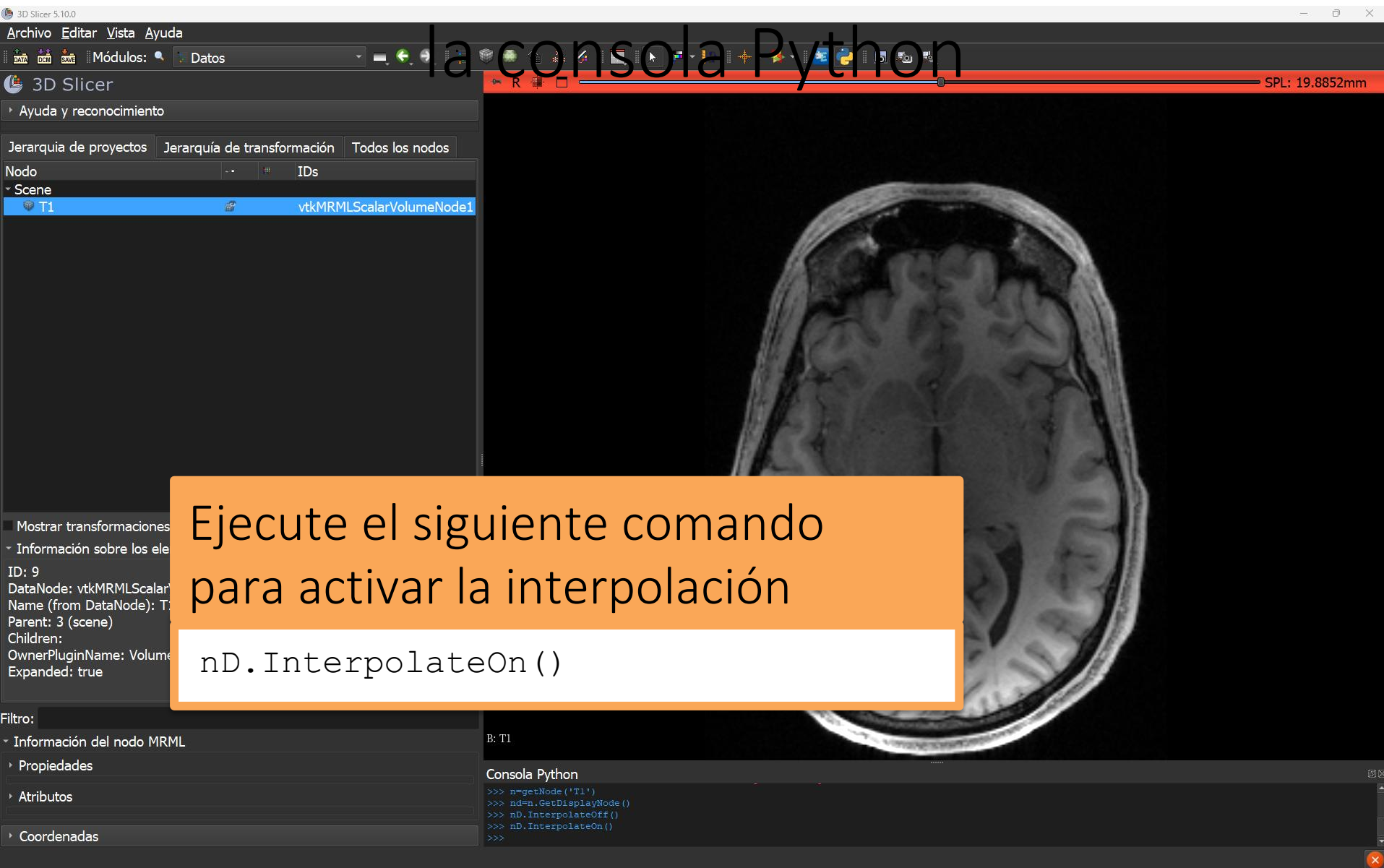
Coordenadas

SPL: 15.8852mm

Slicer muestra la imagen T1 con la interpolación desactivada

```
>>> n=getNode('T1')
>>> nd=n.GetDisplayNode()
>>> nd.InterpolateOff()
>>>
```

Acceso a los nodos MRML desde la consola Python



The screenshot displays the 3D Slicer 5.10.0 interface. The main window shows a brain MRI scan in a sagittal view. The left sidebar contains several panels: 'Ayuda y reconocimiento', 'Jerarquía de proyectos', 'Jerarquía de transformación', and 'Todos los nodos'. The 'Todos los nodos' panel shows a tree structure with 'Scene' expanded, containing 'T1' (vtkMRMLScalarVolumeNode1). The bottom-left panel shows 'Información del nodo MRML' for 'T1', displaying details like ID, DataNode, Name, Parent, Children, OwnerPluginName, and Expanded status. The bottom-right panel is the 'Consola Python' (Python Console), which contains the following code:

```
>>> n=getNode('T1')
>>> nd=n.GetDisplayNode()
>>> nd.InterpolateOff()
>>> nd.InterpolateOn()
>>>
```

An orange text box is overlaid on the center of the image, containing the instruction:

Ejecute el siguiente comando para activar la interpolación

Below this instruction, the code `nd.InterpolateOn()` is shown in a white box.

Acceso a los nodos MRML desde la consola Python

3D Slicer 5.10.0

Archivo Editar Vista Ayuda

Módulos: Datos

3D Slicer

Ayuda y reconocimiento

Jerarquía de proyectos Jerarquía de transformación Todos los nodos

Nodo IDs

Scene

T1 vtkMRMLScalarVolumeNode1

Mostrar transformaciones ▾ Mostrar ID MRML

Información sobre los elementos de la jerarquía temática

ID: 9
DataNode: vtkMRMLScalarVolumeNode1
Name (from DataNode): T1
Parent: 3 (scene)
Children:
OwnerPluginName: Volumes
Expanded: true

Filtro:

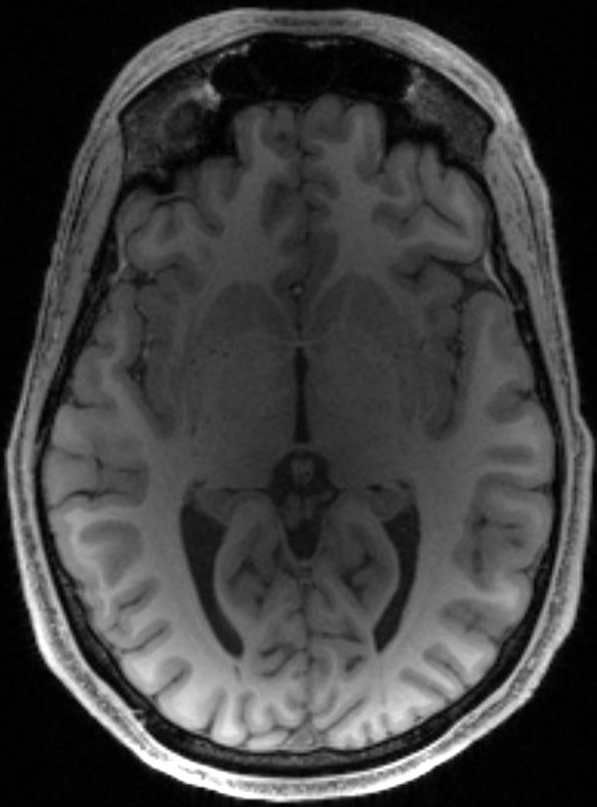
Información del nodo MRML

Propiedades

Atributos

Coordenadas

SPL: 19.8852mm



B: T1

Consola Python

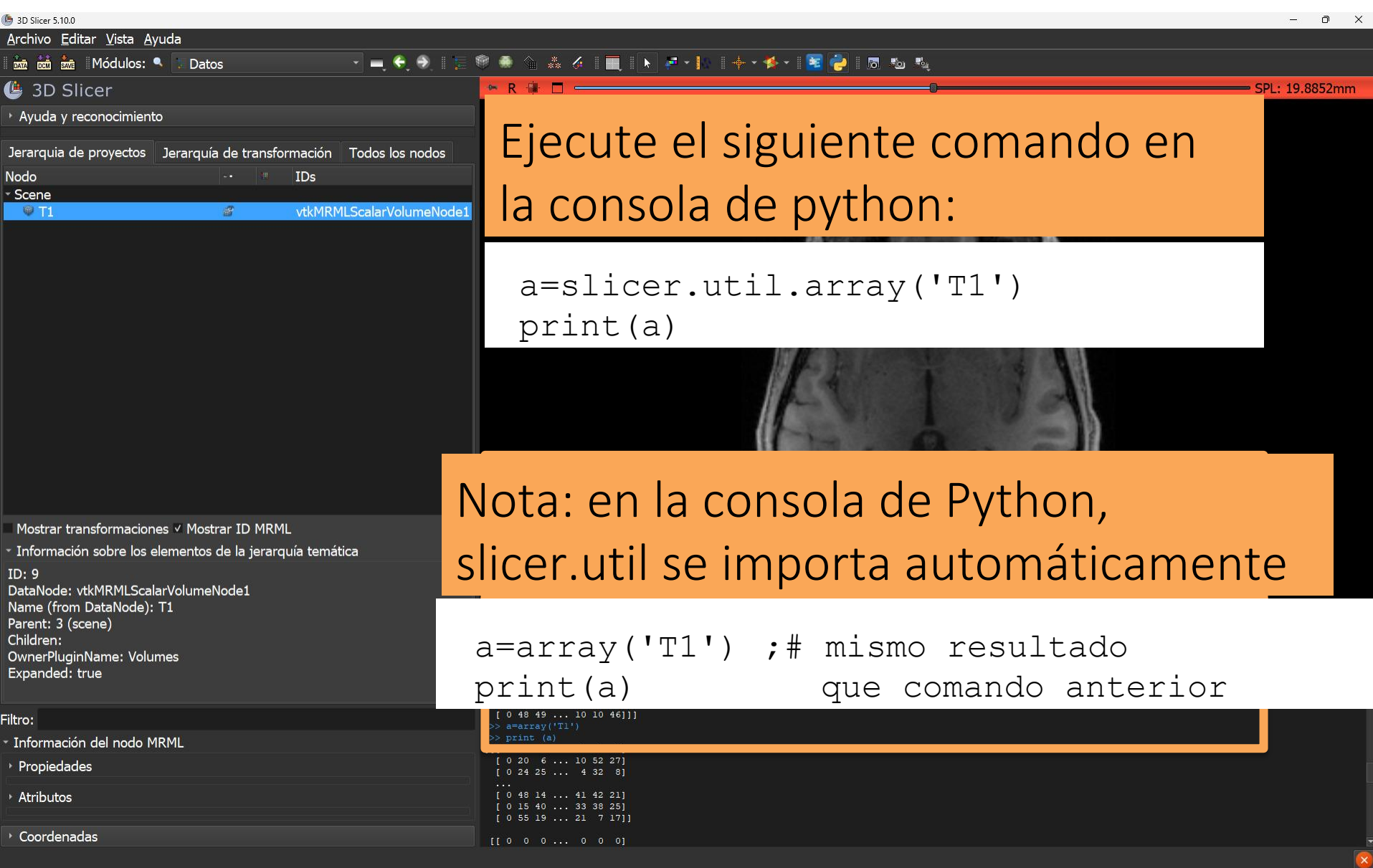
```
>>> n=getNode('T1')
>>> nd=n.GetDisplayNode()
>>> nd.InterpolateOff()
>>> nd.InterpolateOn()
>>>
```

Acceso a los vóxeles de un volumen

- El paquete `slicer.util` da acceso a los volúmenes como matrices multidimensionales NumPy
- Los volúmenes pueden modificarse utilizando métodos estándar de NumPy



Acceso a los vóxeles de un volumen



3D Slicer 5.10.0

Archivo Editar Vista Ayuda

Módulos: Datos

3D Slicer

Ayuda y reconocimiento

Jerarquía de proyectos Jerarquía de transformación Todos los nodos

Nodo IDs

Scene

T1 vtkMRMLScalarVolumeNode1

Mostrar transformaciones ✓ Mostrar ID MRML

Información sobre los elementos de la jerarquía temática

ID: 9
DataNode: vtkMRMLScalarVolumeNode1
Name (from DataNode): T1
Parent: 3 (scene)
Children:
OwnerPluginName: Volumes
Expanded: true

Filtro:

Información del nodo MRML

Propiedades

Atributos

Coordenadas

Ejecute el siguiente comando en la consola de python:

```
a=slicer.util.array('T1')  
print(a)
```

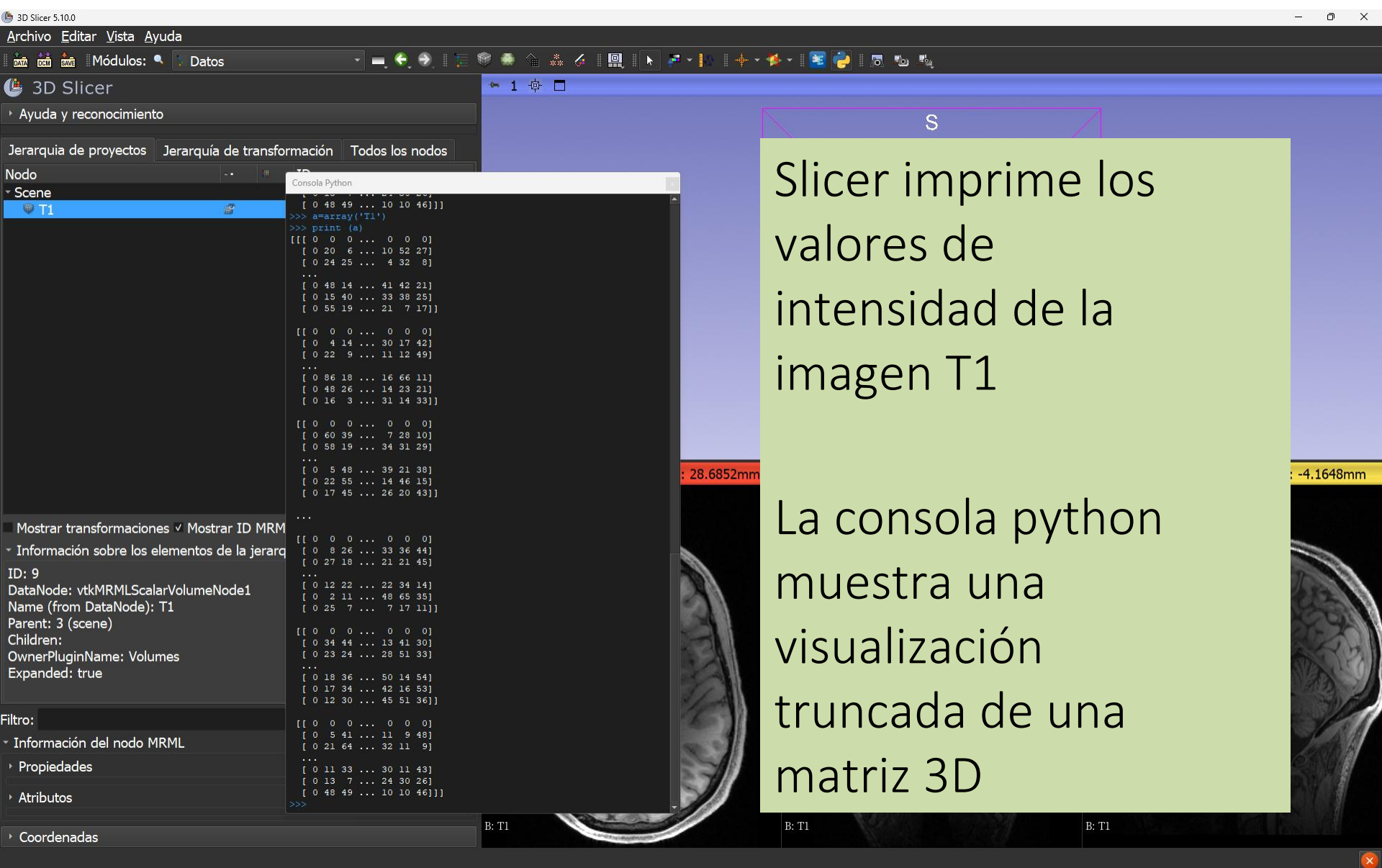
Nota: en la consola de Python, slicer.util se importa automáticamente

```
a=array('T1') ;# mismo resultado  
print(a)           que comando anterior
```

```
[ 0 48 49 ... 10 10 46]]  
>> a=array('T1')  
>> print(a)  
  
[ 0 20 6 ... 10 52 27]  
[ 0 24 25 ... 4 32 8]  
...  
[ 0 48 14 ... 41 42 21]  
[ 0 15 40 ... 33 38 25]  
[ 0 55 19 ... 21 7 17]]  
  
[[ 0 0 0 ... 0 0 0]]
```

SPL: 19.8852mm

Acceso a los vóxeles de un volumen



The screenshot displays the 3D Slicer 5.10.0 interface. The main window shows a T1 MRI volume. A Python console window is open, displaying the following code and output:

```
>>> a=array('T1')
>>> print(a)
[[[ 0  0  0 ...  0  0  0]
 [ 0 20  6 ... 10 52 27]
 [ 0 24 25 ...  4 32  8]
 ...
 [ 0 48 14 ... 41 42 21]
 [ 0 15 40 ... 33 38 25]
 [ 0 55 19 ... 21  7 17]]

[[[ 0  0  0 ...  0  0  0]
 [ 0  4 14 ... 30 17 42]
 [ 0 22  9 ... 11 12 49]
 ...
 [ 0 86 18 ... 16 66 11]
 [ 0 48 26 ... 14 23 21]
 [ 0 16  3 ... 31 14 33]]

[[[ 0  0  0 ...  0  0  0]
 [ 0 60 39 ...  7 28 10]
 [ 0 58 19 ... 34 31 29]
 ...
 [ 0  5 48 ... 39 21 38]
 [ 0 22 55 ... 14 46 15]
 [ 0 17 45 ... 26 20 43]]

...

[[[ 0  0  0 ...  0  0  0]
 [ 0  8 26 ... 33 36 44]
 [ 0 27 18 ... 21 21 45]
 ...
 [ 0 12 22 ... 22 34 14]
 [ 0  2 11 ... 48 65 35]
 [ 0 25  7 ...  7 17 11]]

[[[ 0  0  0 ...  0  0  0]
 [ 0 34 44 ... 13 41 30]
 [ 0 23 24 ... 28 51 33]
 ...
 [ 0 18 36 ... 50 14 54]
 [ 0 17 34 ... 42 16 53]
 [ 0 12 30 ... 45 51 36]]

[[[ 0  0  0 ...  0  0  0]
 [ 0  5 41 ... 11  9 48]
 [ 0 21 64 ... 32 11  9]
 ...
 [ 0 11 33 ... 30 11 43]
 [ 0 13  7 ... 24 30 26]
 [ 0 48 49 ... 10 10 46]]

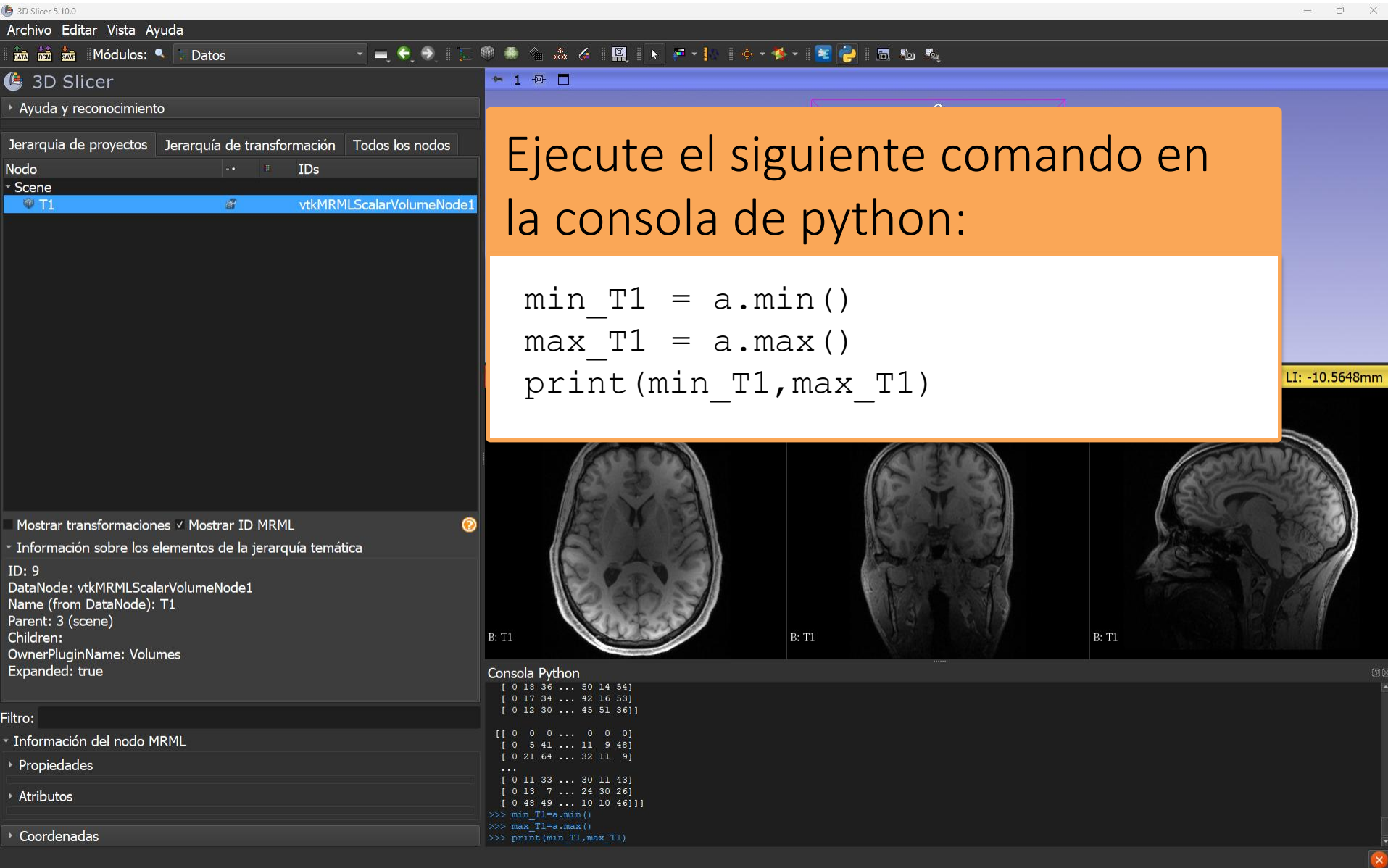
>>>
```

The console output shows a 3D matrix of voxel values. The text overlay on the right indicates that Slicer prints the values of the intensity of the T1 image, and the Python console shows a truncated 3D matrix visualization.

Slicer imprime los valores de intensidad de la imagen T1

La consola python muestra una visualización truncada de una matriz 3D

Acceso a los vóxeles de un volumen



The screenshot displays the 3D Slicer 5.10.0 interface. On the left, the 'Scene' panel shows a T1 volume node. The bottom-left panel shows the 'Information about the thematic hierarchy elements' section, which includes details about the T1 volume node. The main 3D view shows three orthogonal slices of the T1 volume. An orange box is overlaid on the 3D view, containing a command to execute in the Python console. The Python console at the bottom shows the execution of the command, resulting in the minimum and maximum values of the volume.

Ejecute el siguiente comando en la consola de python:

```
min_T1 = a.min()
max_T1 = a.max()
print(min_T1,max_T1)
```

Mostrar transformaciones ✓ Mostrar ID MRML

Información sobre los elementos de la jerarquía temática

ID: 9
DataNode: vtkMRMLScalarVolumeNode1
Name (from DataNode): T1
Parent: 3 (scene)
Children:
OwnerPluginName: Volumes
Expanded: true

Filtro:

Información del nodo MRML

Propiedades

Atributos

Coordenadas

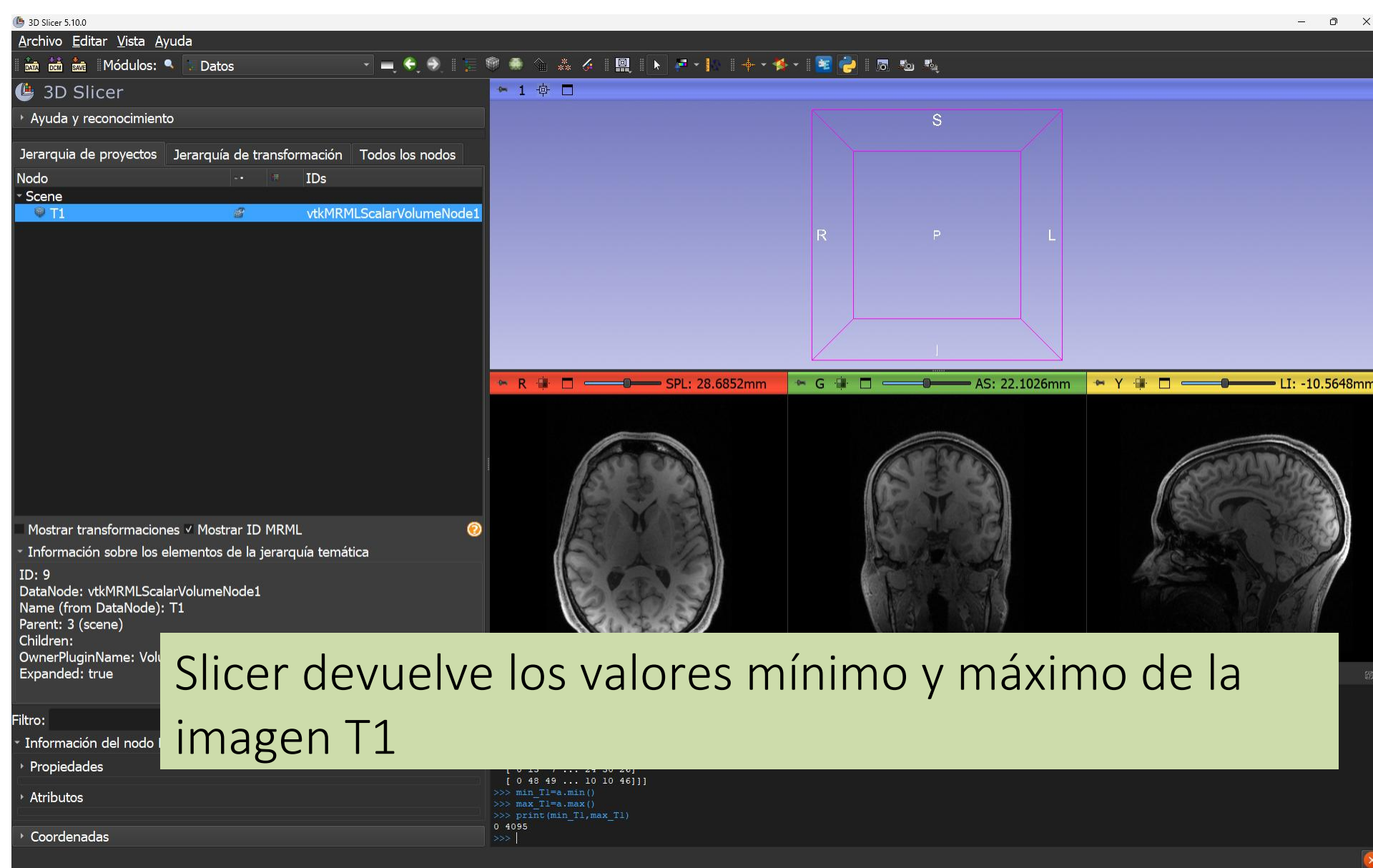
Console Python

```
[ 0 18 36 ... 50 14 54]
[ 0 17 34 ... 42 16 53]
[ 0 12 30 ... 45 51 36]]

[[ 0 0 0 ... 0 0 0]
 [ 0 5 41 ... 11 9 48]
 [ 0 21 64 ... 32 11 9]
 ...
 [ 0 11 33 ... 30 11 43]
 [ 0 13 7 ... 24 30 26]
 [ 0 48 49 ... 10 10 46]]]

>>> min_T1=a.min()
>>> max_T1=a.max()
>>> print(min_T1,max_T1)
```

Acceso a los vóxeles de un volumen



The screenshot displays the 3D Slicer 5.10.0 interface. The top menu bar includes 'Archivo', 'Editar', 'Vista', and 'Ayuda'. Below it is a toolbar with various icons. The left sidebar contains a '3D Slicer' header, a 'Módulos:' dropdown, and a 'Datos' button. The main panel shows a 3D view of a T1 MRI volume with a purple wireframe box indicating the current slice. The bottom panel shows three orthogonal views (axial, sagittal, and coronal) of the same volume. A green box highlights the text 'Slicer devuelve los valores mínimo y máximo de la imagen T1'.

3D Slicer 5.10.0

Archivo Editar Vista Ayuda

Módulos: Datos

3D Slicer

Ayuda y reconocimiento

Jerarquía de proyectos Jerarquía de transformación Todos los nodos

Nodo

Scene

T1 vtkMRMLScalarVolumeNode1

Mostrar transformaciones ✓ Mostrar ID MRML

Información sobre los elementos de la jerarquía temática

ID: 9

DataNode: vtkMRMLScalarVolumeNode1

Name (from DataNode): T1

Parent: 3 (scene)

Children:

OwnerPluginName: Vol

Expanded: true

Filtro:

Información del nodo

Propiedades

Atributos

Coordenadas

Slicer devuelve los valores mínimo y máximo de la imagen T1

```
[ 0 49 7 11 21 30 40]
[ 0 48 49 ... 10 10 46]]
>>> min_T1=a.min()
>>> max_T1=a.max()
>>> print(min_T1,max_T1)
0 4095
>>> |
```

Modificación de vóxeles en un volumen

3D Slicer 5.10.0

Archivo Editar Vista Ayuda

DATA DATA SAVE Módulos: Datos

3D Slicer

Ayuda y reconocimiento

Jerarquía de proyectos Jerarquía de transformación Todos los nodos

Nodo IDs

Scene

T1 vtkMRMLScalarVolumeNode1

Mostrar transformaciones ✓ Mostrar ID MRML

Información sobre los elementos de la jerarquía temática

ID: 9
DataNode: vtkMRMLScalarVolumeNode1
Name (from DataNode): T1
Parent: 3 (scene)
Children:
OwnerPluginName: Volumes
Expanded: true

Filtro:

Información del nodo MRML

Propiedades

Atributos

Coordenadas

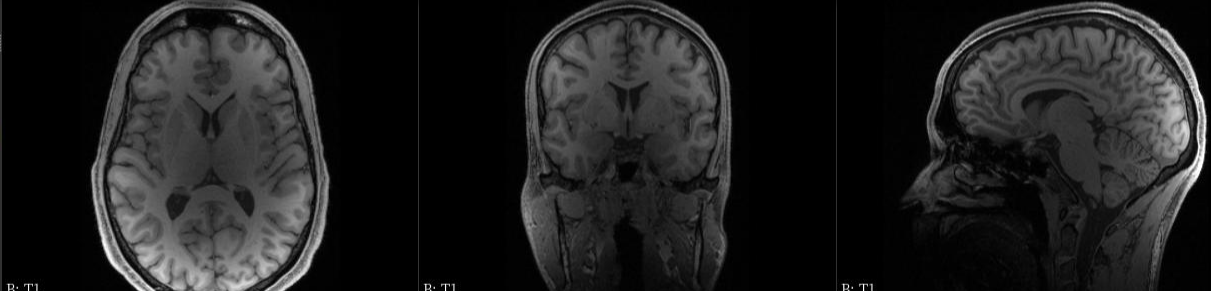
Ejecute el siguiente comando en la consola de python:

```
print(a.shape)
```

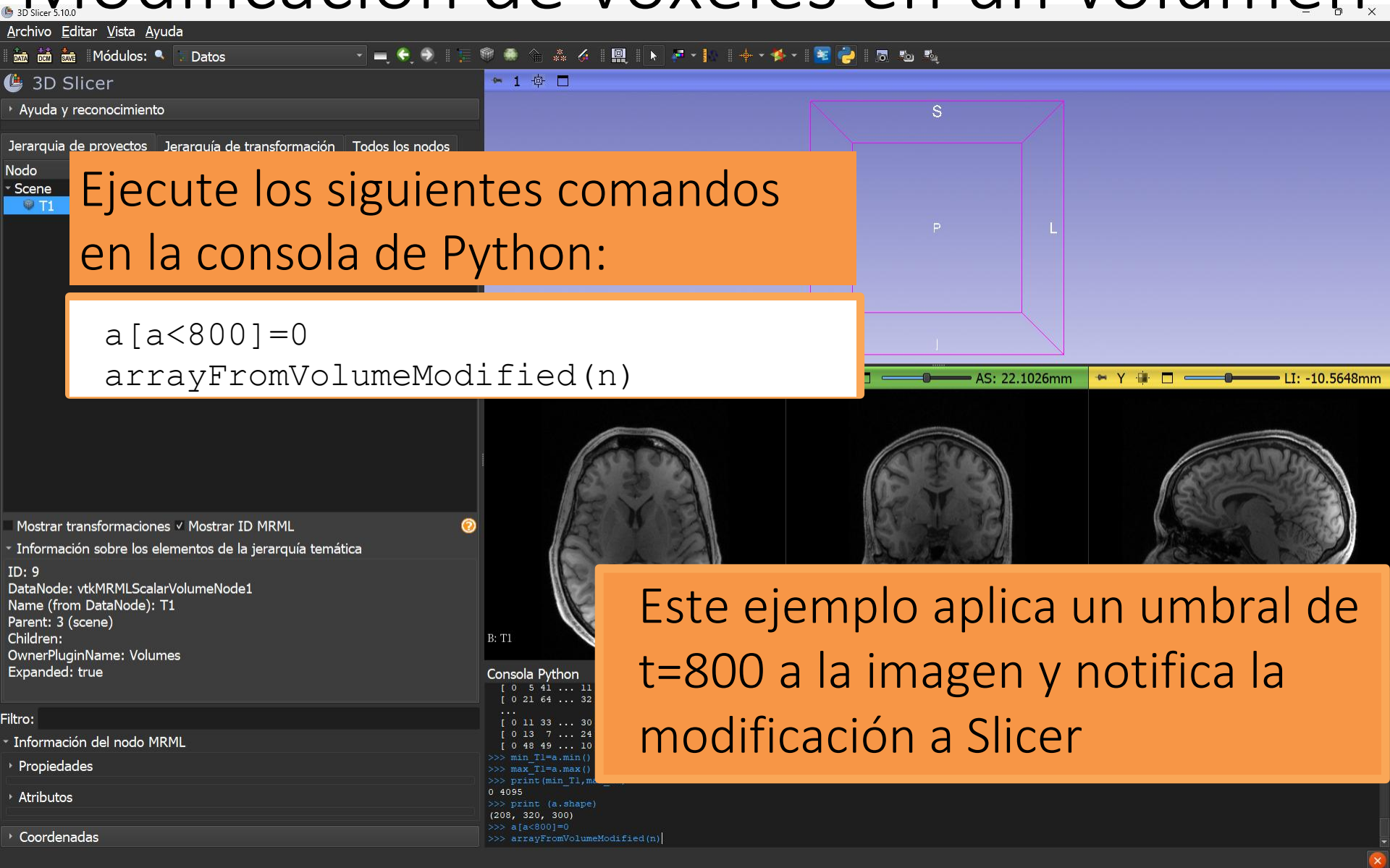
LI: -10.5648mm

Slicer devuelve las dimensiones de la imagen

```
[ 0 11 33 ... 30 11 43]  
[ 0 13 7 ... 24 30 26]  
[ 0 48 49 ... 10 10 46]]  
  
>>> min_T1=a.min()  
>>> max_T1=a.max()  
>>> print(min_T1,max_T1)  
0 4095  
>>> print(a.shape)  
(208, 320, 300)  
>>>
```



Modificación de vóxeles en un volumen



Ejecute los siguientes comandos en la consola de Python:

```
a[a<800]=0  
arrayFromVolumeModified(n)
```

Este ejemplo aplica un umbral de $t=800$ a la imagen y notifica la modificación a Slicer

Mostrar transformaciones ☒ Mostrar ID MRML

Información sobre los elementos de la jerarquía temática

ID: 9
DataNode: vtkMRMLScalarVolumeNode1
Name (from DataNode): T1
Parent: 3 (scene)
Children:
OwnerPluginName: Volumes
Expanded: true

Filtro:

Información del nodo MRML

Propiedades

Atributos

Coordenadas

B: T1

Consola Python

```
[ 0  5 41 ... 11  
[ 0 21 64 ... 32  
...  
[ 0 11 33 ... 30  
[ 0 13  7 ... 24  
[ 0 48 49 ... 10  
...  
>>> min_T1=a.min()  
>>> max_T1=a.max()  
>>> print(min_T1,max_T1)  
0 4095  
>>> print (a.shape)  
(208, 320, 300)  
>>> a[a<800]=0  
>>> arrayFromVolumeModified(n)
```

AS: 22.1026mm Y LI: -10.5648mm

Modificación de vóxeles en un volumen

3D Slicer 5.10.0

Archivo Editar Vista Ayuda

Módulos: Datos

3D Slicer

Ayuda y reconocimiento

Jerarquía de proyectos Jerarquía de transformación Todos los nodos

Nodo IDs

Scene

T1 vtkMRMLScalarVolumeNode1

Mostrar transformaciones ☒ Mostrar ID MRML

Información sobre los elementos de la jerarquía temática

ID: 9
DataNode: vtkMRMLScalarVolumeNode1
Name (from DataNode): T1
Parent: 3 (scene)
Children:
OwnerPluginName: Volumes
Expanded: true

Filtro:

Información del nodo MRML

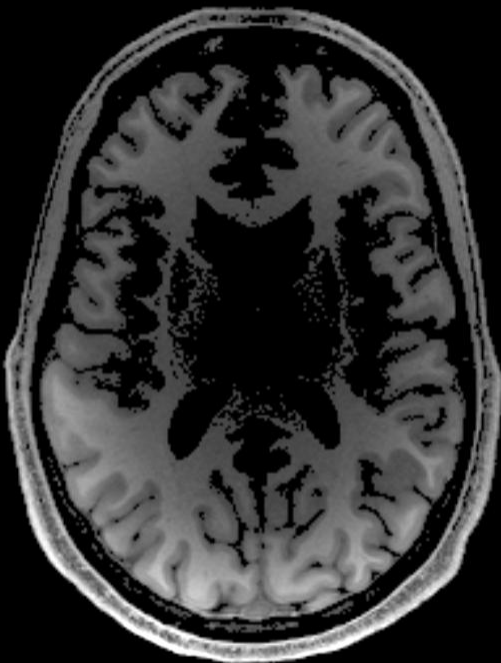
Propiedades

Atributos

Coordenadas

R

SPL: 33.4852mm



B: T1

Consola Python

```
[ 0 21 64 ... 32 11 9]  
...  
[ 0 11 33 ... 30 11 43]  
[ 0 13 7 ... 24 30 26]  
[ 0 48 49 ... 10 10 46]]]  
>>> min_T1=a.min()  
>>> max_T1=a.max()  
>>> print(min_T1,max_T1)  
0 4095  
>>> print (a.shape)  
(208, 320, 300)  
>>> a[a<800]=0  
>>> arrayFromVolumeModified(n)  
>>> |
```

Slicer muestra el resultado del umbral

Cargar el volumen T2



3D Slicer 5.10.0

Archivo Editar Vista Ayuda

Módulos: Datos

3D Slicer

Ayuda y reconocimiento

Jerarquía

Nodo

Scene

T1

T2

vtkMRMLScalarVolumeNode2

SPL: 2.2852mm

Cargue la imagen T2 en Slicer:

```
slicer.util.loadVolume ('C:/Users/SlicerProgrammingTutorialData/T2.nrrd')
```

Consola Python

```
SyntaxError: (unicode error) 'unicodeescape' codes can't decode bytes in position 2-3: truncated \U0000000X escape  
>>> slicer.util.loadVolume('C:/Users/AVilchisG/Dropbox/AV/3DSlicer_projet_sabatico/tercer año/Tutoriales_FW_laspalmas/SlicerProgrammingTutorialData/T2.nrrd')  
<MRMLCorePython.vtkMRMLScalarVolumeNode (0x000001F73E3275C0) at 0x000001F7359FEF80>  
>>>
```

Mostrar transformaciones ✓ Mostrar ID MRML

Información sobre los elementos de la jerarquía temática

ID: 9
DataNode: vtkMRMLScalarVolumeNode1
Name (from DataNode): T1
Parent: 3 (scene)
Children:
OwnerPluginName: Volumes
Expanded: true

Filtro:

Información del nodo MRML

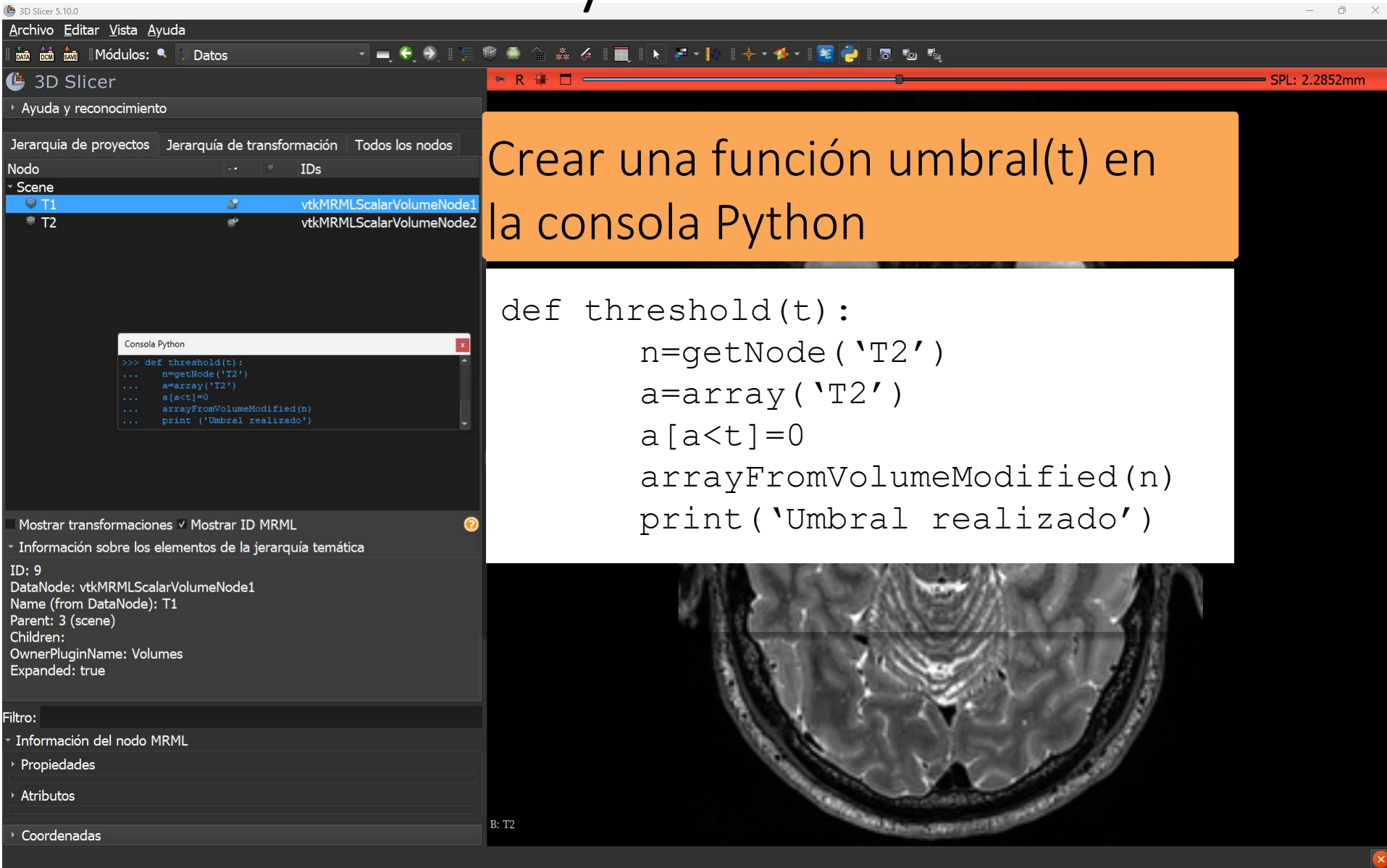
Propiedades

Atributos

Coordenadas

B: T2

Función Python: umbral



The screenshot displays the 3D Slicer 5.10.0 software interface. On the left, the 'Ayuda y reconocimiento' sidebar is visible, showing a hierarchy of nodes with 'T1' and 'T2' selected. Below this, the 'Consola Python' window shows the following code:

```
>>> def threshold(t):  
...     n=getNode('T2')  
...     a=array('T2')  
...     a[a<t]=0  
...     arrayFromVolumeModified(n)  
...     print('Umbral realizado')
```

Below the console, the 'Mostrar transformaciones' and 'Mostrar ID MRML' sections are visible, along with a detailed view of the selected node 'T1'.

On the right, a large orange box contains the text: 'Crear una función umbral(t) en la consola Python'.

Below the orange box, a white box contains the Python code for the threshold function:

```
def threshold(t):  
    n=getNode('T2')  
    a=array('T2')  
    a[a<t]=0  
    arrayFromVolumeModified(n)  
    print('Umbral realizado')
```

At the bottom right, a 3D visualization of a brain MRI scan is shown, labeled 'B: T2'.

Función Python: umbral

3D Slicer 5.10.0

Archivo Editar Vista Ayuda

Módulos: Datos

3D Slicer

Ayuda y reconocimiento

Jerarquía de proyectos Jerarquía de transformación Todos los nodos

Nodo IDs

Scene

T1 vtkMRMLScalarVolumeNode1

T2

Ejecuta la función umbral:
threshold(500)

```
... print ('Umbral realizado')  
...  
>>> threshold(500)
```

Mostrar transformaciones ✓ Mostrar ID MRML

Información sobre los elementos de la jerarquía temática

ID: 9
DataNode: vtkMRMLScalarVolumeNode1
Name (from DataNode): T1
Parent: 3 (scene)
Children:
OwnerPluginName: Volumes
Expanded: true

Filtro:

Información del nodo MRML

Propiedades

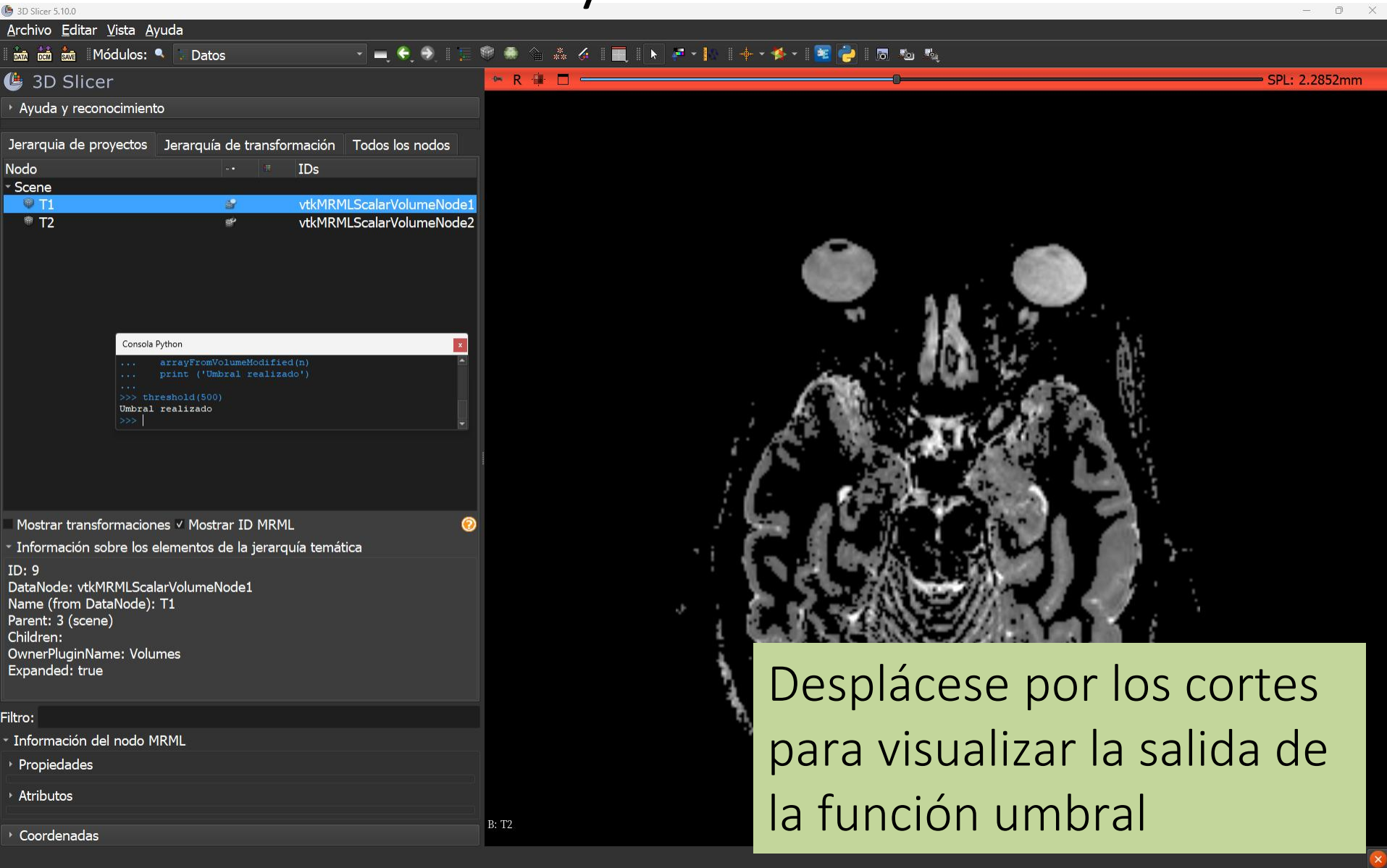
Atributos

Coordenadas

SPL: 2.2852mm

B: T2

Función Python: umbral



The screenshot displays the 3D Slicer 5.10.0 interface. The main window shows a coronal slice of a brain MRI volume. The left sidebar contains the 'Scene' panel with a tree view showing 'T1' and 'T2' volumes. The 'Console Python' panel is open, showing a script that applies a threshold of 500 to the selected volume. The bottom status bar indicates the current slice is 'B: T2'.

3D Slicer 5.10.0

Archivo Editar Vista Ayuda

Módulos: Datos

3D Slicer

Ayuda y reconocimiento

Jerarquía de proyectos Jerarquía de transformación Todos los nodos

Nodo IDs

Scene

T1 vtkMRMLScalarVolumeNode1

T2 vtkMRMLScalarVolumeNode2

Console Python

```
... arrayFromVolumeModified(m)
... print ('Umbral realizado')
...
>>> threshold(500)
Umbral realizado
>>> |
```

Mostrar transformaciones ✓ Mostrar ID MRML

Información sobre los elementos de la jerarquía temática

ID: 9

DataNode: vtkMRMLScalarVolumeNode1

Name (from DataNode): T1

Parent: 3 (scene)

Children:

OwnerPluginName: Volumes

Expanded: true

Filtro:

Información del nodo MRML

Propiedades

Atributos

Coordenadas

SPL: 2.2852mm

B: T2

Desplácese por los cortes para visualizar la salida de la función umbral

Panaroma General

- Slicer facilita el acceso para analizar y modificar tipos de datos complejos
- Slicer es compatible con una amplia gama de paquetes informáticos científicos de Python
- Slicer es un entorno de investigación para realizar experimentos con imágenes médicas

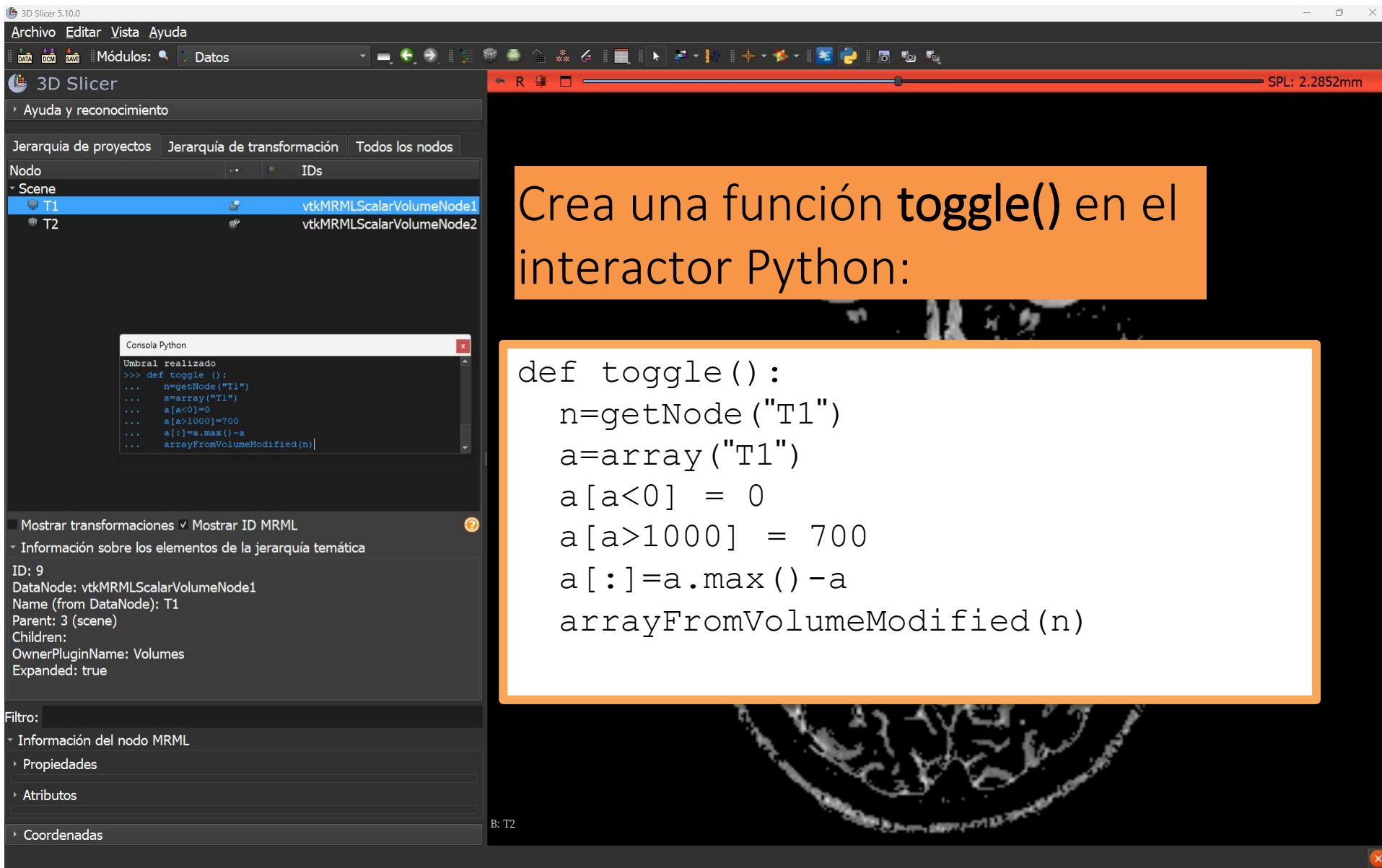
Parte 3

Familiarizarse con Qt en Slicer

Qt & PythonQt

- **Qt** es la herramienta principal de Slicer para crear widgets, cuadros de diálogo, entradas de texto, etc.
- **PythonQt** usa la mayoría de las funcionalidades de Qt y es accesible a través del interactuador Python en Slicer.
- Las interfaces de usuario se pueden crear sobre la marcha para una rápida creación de prototipos y depuración.

Función Python: toggle



The screenshot shows the 3D Slicer 5.10.0 interface. The top menu bar includes 'Archivo', 'Editar', 'Vista', and 'Ayuda'. Below it is a toolbar with various icons. The left sidebar contains a 'Jerarquía de proyectos' panel with a tree view showing 'Scene' containing 'T1' (vtkMRMLScalarVolumeNode1) and 'T2' (vtkMRMLScalarVolumeNode2). Below this is a 'Consola Python' window with the following code:

```
Umbral realizado
>>> def toggle():
...     n=getNode("T1")
...     a=array("T1")
...     a[a<0]=0
...     a[a>1000]=700
...     a[:]=a.max()-a
...     arrayFromVolumeModified(n)
```

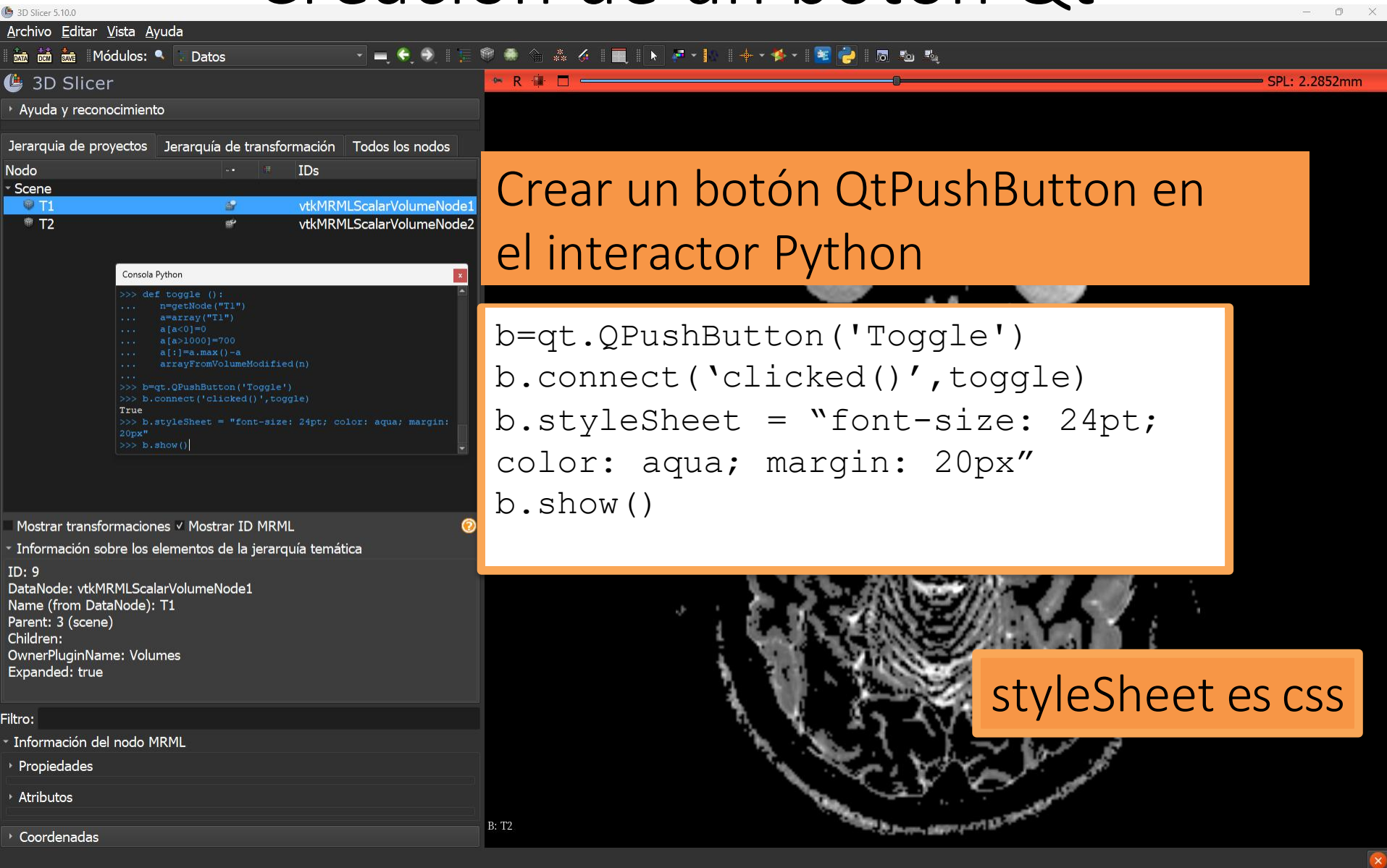
Below the console is a panel showing 'Mostrar transformaciones' and 'Mostrar ID MRML'. The 'Información sobre los elementos de la jerarquía temática' section displays details for 'ID: 9', 'DataNode: vtkMRMLScalarVolumeNode1', 'Name (from DataNode): T1', 'Parent: 3 (scene)', 'Children:', 'OwnerPluginName: Volumes', and 'Expanded: true'. The 'Filtro:' section shows 'Información del nodo MRML', 'Propiedades', 'Atributos', and 'Coordenadas'. The main 3D view shows a brain scan with a red overlay. The status bar at the bottom right indicates 'SPL: 2.2852mm'.

Crea una función **toggle()** en el interactor Python:

```
def toggle():
    n=getNode("T1")
    a=array("T1")
    a[a<0] = 0
    a[a>1000] = 700
    a[:]=a.max()-a
    arrayFromVolumeModified(n)
```

B: T2

Creación de un botón Qt



The screenshot shows the 3D Slicer 5.10.0 interface. The left sidebar contains the 'Jerarquía de proyectos' (Project Hierarchy) panel, which lists two nodes: 'T1' (vtkMRMLScalarVolumeNode1) and 'T2' (vtkMRMLScalarVolumeNode2). Below this is the 'Console Python' window, which displays the following code:

```
>>> def toggle():
...     n=getNode("T1")
...     a=array("T1")
...     a[a<0]=0
...     a[a>1000]=700
...     a[:]=a.max()-a
...     arrayFromVolumeModified(n)
...
>>> b=qt.QPushButton('Toggle')
>>> b.connect('clicked()',toggle)
True
>>> b.setStyleSheet = "font-size: 24pt; color: aqua; margin: 20px"
>>> b.show()
```

The right sidebar contains the 'Mostrar transformaciones' (Show Transformations) panel, which shows the 'Información sobre los elementos de la jerarquía temática' (Information about the thematic hierarchy elements) for node 'T1'. The information includes: ID: 9, DataNode: vtkMRMLScalarVolumeNode1, Name (from DataNode): T1, Parent: 3 (scene), Children: (empty), OwnerPluginName: Volumes, and Expanded: true.

The main 3D view displays a cross-sectional image of a brain. The status bar at the bottom right indicates 'SPL: 2.2852mm'.

Crear un botón QPushButton en el interactor Python

```
b=qt.QPushButton('Toggle')
b.connect('clicked()',toggle)
b.setStyleSheet = "font-size: 24pt;
color: aqua; margin: 20px"
b.show()
```

styleSheet es css

Creación de un botón Qt

The image shows the 3D Slicer 5.10.0 interface. On the left, the 'Console Python' window displays the following code:

```
... n=getNode("T1")
... a=array("T1")
... a[a<0]=0
... a[a>1000]=700
... a[:]=a.max()-a
... arrayFromVolumeModified(n)
...
>>> b=qt.QPushButton('Toggle')
>>> b.connect('clicked()',toggle)
True
>>> b.setStyleSheet = "font-size: 24pt; color: aqua; margin: 20px"
>>> b.show()
>>>
```

Below the console, the 'Mostrar transformaciones' and 'Mostrar ID MRML' checkboxes are checked. The 'Información sobre los elementos de la jerarquía temática' section shows the following details for the selected node:

ID: 9
DataNode: vtkMRMLScalarVolumeNode1
Name (from DataNode): T1
Parent: 3 (scene)
Children:
OwnerPluginName: Volumes
Expanded: true

The 'Filtro:' section shows 'Información del nodo MRML' expanded, with 'Propiedades' and 'Atributos' sub-sections visible.

On the right, the 3D Slicer main window displays a brain scan. A small window titled 'Slicer' is overlaid on the scan, showing a button labeled 'Toggle' in aqua text. The button is styled with a font size of 24pt and a margin of 20px.

A green banner at the bottom right of the image contains the text: **Aparece el botón Alternar**

Creación de un botón Qt

The screenshot displays the 3D Slicer 5.10.0 interface. On the left, the 'Hierarchy' panel shows a tree structure with 'Scene' as the root, containing 'T1' and 'T2' nodes. 'T2' is selected, and its details are shown in the 'Properties' panel below. The 'Console Python' window shows a script snippet:

```
n=getNode("T1")  
array("T1")
```

. A red arrow points from the 'T1' node in the hierarchy to the 'Console Python' window. An orange callout box with the text 'Activar la visibilidad del volumen T1' is positioned over the 'Console Python' window. In the center, a small Qt window titled 'Slicer' contains a button labeled 'Toggle'. The main 3D view on the right shows a grayscale axial brain slice. The status bar at the bottom right indicates 'SPL: 11.8852mm'.

3D Slicer 5.10.0

Archivo Editar Vista Ayuda

Módulos: Datos

3D Slicer

Ayuda y reconocimiento

Jerarquía de proyectos Jerarquía de transformación Todos los nodos

Nodo IDs

Scene

T1 vtkMRMLScalarVolumeNode1

T2 vtkMRMLScalarVolumeNode2

Console Python

```
n=getNode("T1")  
array("T1")
```

Activar la visibilidad del volumen T1

Mostrar transformaciones Mostrar ID de nodo

Información sobre los elementos de la jerarquía temática

ID: 12

DataNode: vtkMRMLScalarVolumeNode2

Name (from DataNode): T2

Parent: 3 (scene)

Children:

OwnerPluginName: Volumes

Expanded: true

Filtro:

Información del nodo MRML

Propiedades

Atributos

Coordenadas

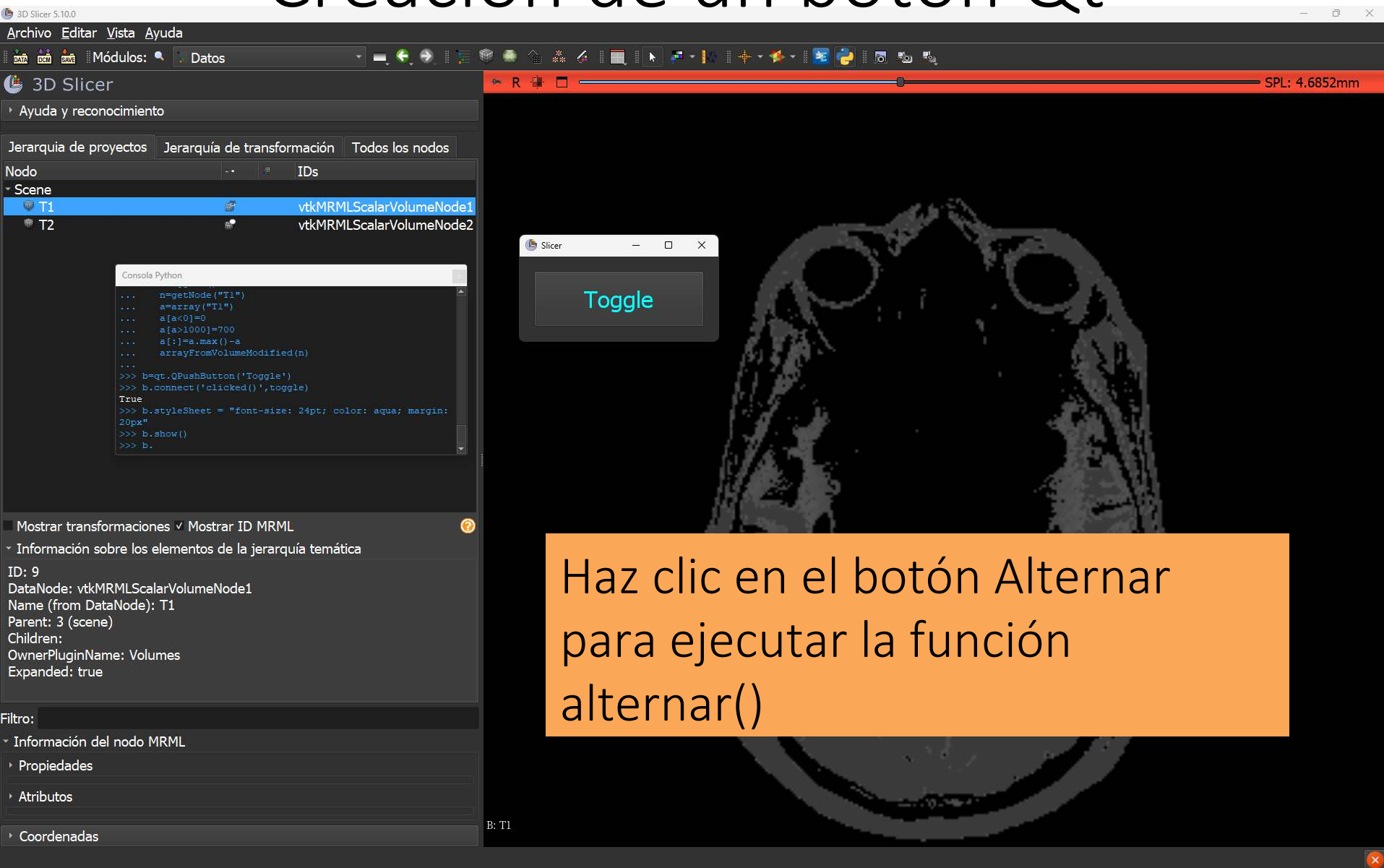
Slicer

Toggle

B: T2

SPL: 11.8852mm

Creación de un botón Qt



The image shows the 3D Slicer 5.10.0 interface. On the left, the 'Nodo' panel lists 'T1' and 'T2' under the 'Scene' node. The 'Console Python' window displays the following code:

```
... n=getNode("T1")
... a=array("T1")
... a[a<0]=0
... a[a>1000]=700
... a[:]=a.max()-a
... arrayFromVolumeModified(m)
...
>>> b=qt.QPushButton('Toggle')
>>> b.connect('clicked()',toggle)
True
>>> b.setStyleSheet = "font-size: 24pt; color: aqua; margin: 20px"
>>> b.show()
>>> b.
```

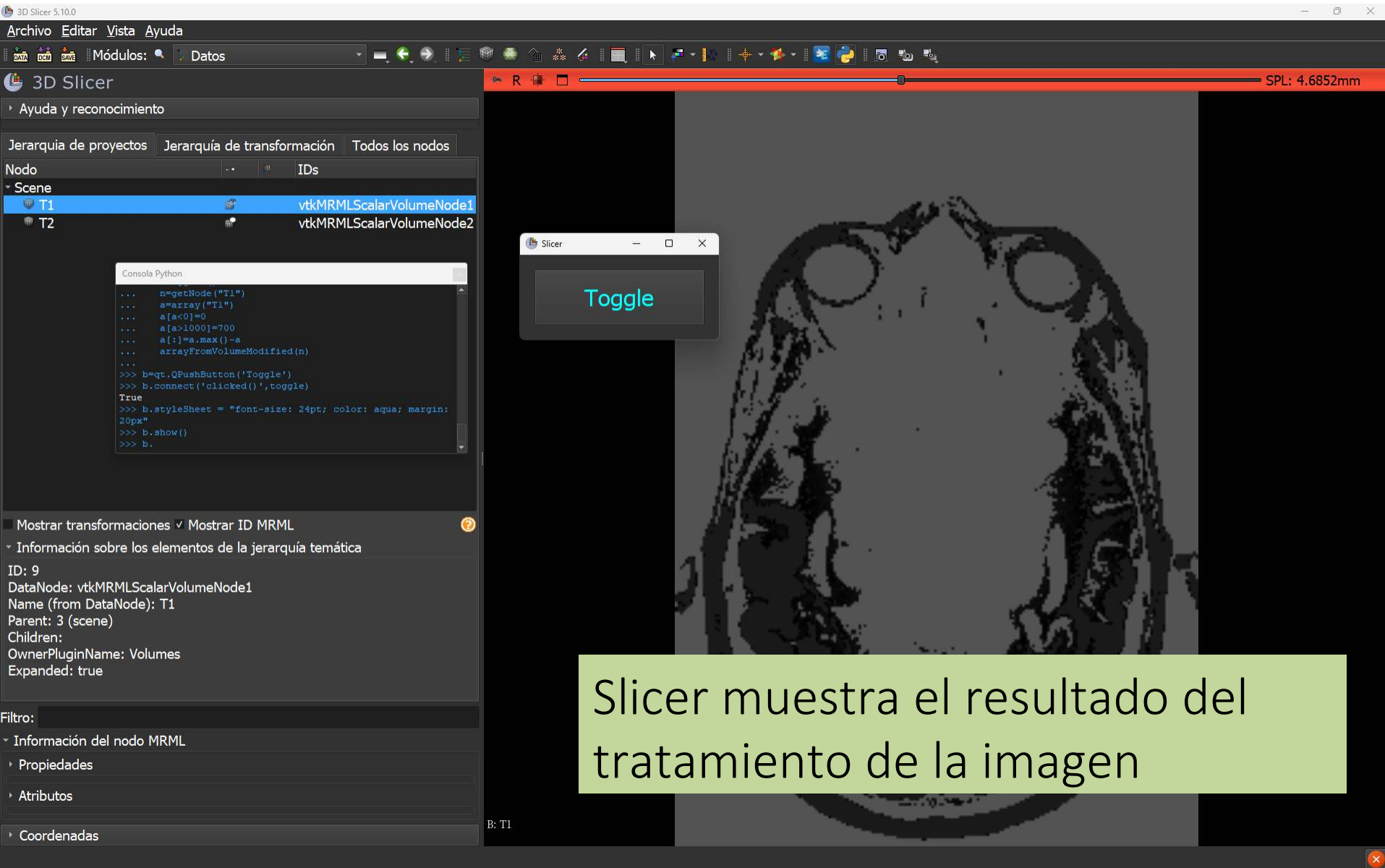
Below the console, the 'Información sobre los elementos de la jerarquía temática' panel shows details for 'T1' (DataNode: vtkMRMLScalarVolumeNode1, Name: T1, Parent: 3 (scene), Children: 0, OwnerPluginName: Volumes, Expanded: true).

On the right, a large MRI scan is displayed. A small window titled 'Slicer' is overlaid on the scan, containing a button labeled 'Toggle' in cyan text. The button is positioned over the top part of the scan, near the eyes.

At the bottom right, an orange box contains the text:

Haz clic en el botón Alternar para ejecutar la función alternar()

Creación de un botón Qt



Ejemplos de módulos con secuencias de comandos

- El tutorial muestra cómo crear una interfaz sencilla en Python.
- Slicer integra muchos módulos de secuencias de comandos sofisticados, como estadísticas de segmentos, datos de muestras, módulo de endoscopia, etc.
- Para más información, consulte el repositorio de secuencias de comandos de Slicer:

<https://www.slicer.org/wiki/Documentation/Nightly/ScriptRepository>

Conclusión

- Slicer permite a los desarrolladores crear interfaces complejas y ágiles para los usuarios
- La plataforma de software ofrece posibilidades de personalización ilimitadas
- Slicer da acceso a bibliotecas subyacentes avanzadas a través de un paquete multiplataforma que es fácil de desplegar a los usuarios

Agradecimientos



Neuroimage Analysis Center
(NIBIB P41 EB015902)